



## INDEX

- 1 E-CELL2の特徴**
- 2 E-CELL2.25のデモンストレーション**
- 3 E-CELL2のルール作成**
  - 3.1 スプレッドシート作成
  - 3.2 ルールファイルとルール・インターメディエイト・ファイル作成
- 4 E-CELL2のユーザー定義Reactorの作成**
  - 4.1 Reactorの概略
  - 4.2 Reactorの作成手順
- 5 E-CELL2の操作方法**

## 1 E-CELL2の特徴

この章では、E-CELL2.25の特徴と操作方法について、概略を説明していきます。E-CELLの理論的背景についての詳細は、the E-CELL.Org(<http://www.e-cell.org>)で配布(予定)のチュートリアルを参照して下さい。

細胞をモデル化するためには、構成要素の抽象化が必要になってきます。E-CELLでは、細胞の構造と、細胞内の化学反応を表現するために、“Substance-Reactor Model”を採用しています。このモデルには、以下のような特徴があります。

- 細胞の状態を「物質(Substance)の量」の集合として捉える。
- 細胞の活動は化学反応式(Reactor)による「Substanceの量の変化」で表される。
- Substanceの複数のノードが、Reactorによって連結される有向グラフになる。

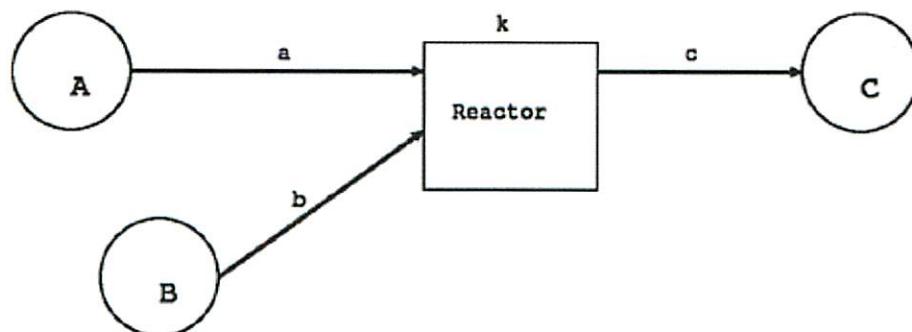


図1. Substance-Reactor Model

- 物質の局在(System)を用いて、細胞内小器官を表現する。

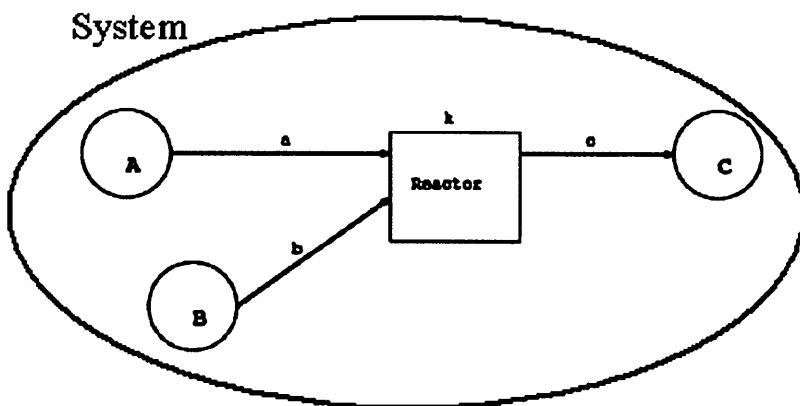


図2. Systemを考慮したSubstance-Reactor Model

ここで注意しなければならないのは、物質(Substance)という概念が、通常の化学分子、イオン、ラジカルのみならず、浸透圧や細胞の体積といった、任意の物理量まで含んでいることです。但し、負の値は表現できません。

E-CELLでは、細胞内が均一であることを仮定しています。従って、拡散を取り扱うには、物質の量が異なる、複数のシステムを用いる必要があります。

E-CELLでは細胞内の様々な現象をシミュレーションするために、酵素反応のような反応速度(流束)に基づく反応と化学平衡のような一瞬で起こる反応を混在させることができます。

まず、流束に基づく(微分方程式として解ける)化学反応は、E-CELL内部では 次のようにして扱われます。

- 通常(Regular) Reactorを使用して、化学反応式の反応速度を変更する。
- ルンゲクッタ法を用いて、常微分方程式を解く。
- 代表的な反応速度式としては、Michaelis-Menten反応が挙げられる。

化学平衡等の(代数方程式として解ける)化学反応については、次のようにして扱われます。

- 裏口(Postern) Reactorを使用して、Substanceの量を直接操作する。
- 迅速平衡や浸透圧の変化などがこのタイプの反応に当たる。

ソフトウェア工学的な見地からは、E-CELL2は以下のようない特徴を持っています。まず、C++言語で記述されている部分がどうなっているかを見てみましょう。この部分はE-CELL2の心臓にあたり、E-CELL2が細胞をどう表現しているかを示す、世界観(Ontology)に相当します。構成部品はオブジェクト指向モデリングの用語ではクラスと呼ばれ、各々のクラスが作用しあって、シミュレーションを行うための計算機構が実現されます。

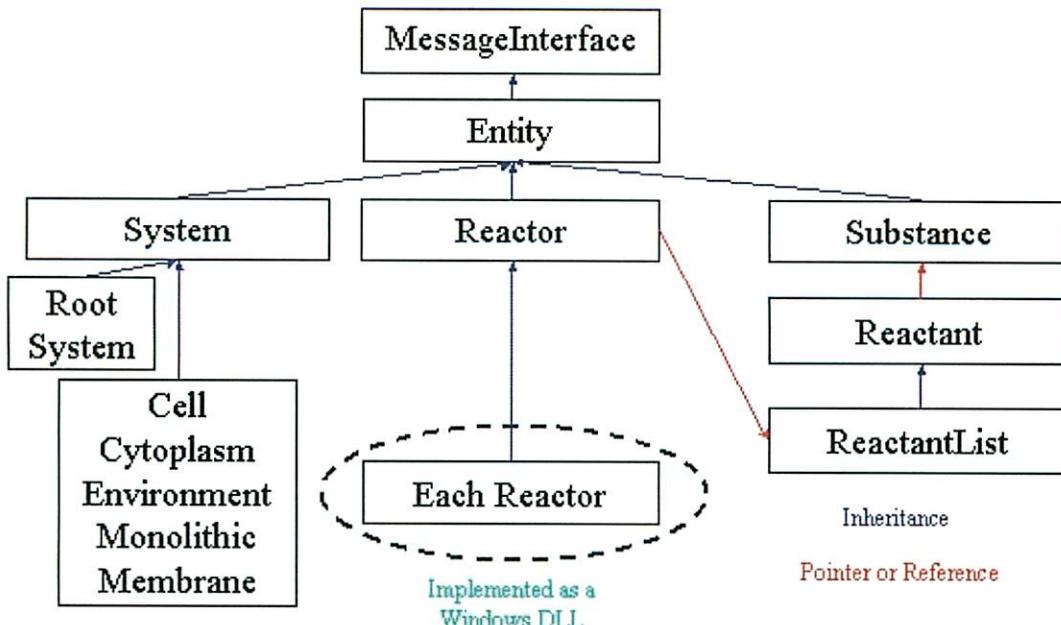


図3. E-CELL2の内部構成

Linux用に開発された、E-CELL1から変更された点を挙げておきましょう。

- 多重継承の廃止
- フレンド関数廃止
- AccumulatorクラスおよびIntegratorクラスのSubstanceクラスへの統合
- StepperクラスのSystemクラスへの統合
- モデリング・ランチャーの実装

これらの処理は、Linux上で動作していたE-CELL1の移植性を高めるために行われました。一方、GUI (Graphic User Interface) や、E-CELL2の操作を自動化するための、スクリプトファイルを解釈する部分や、モデリング・ランチャーは、Java言語で記述されています。

E-CELL2には、対話的に操作可能なGUI版と、コマンドラインから実行可能なバッチ版があります。2章では、GUI版についてのみ説明を行います。バッチ版の説明は3章2.9節“E-CELL2 バッチ版の使用とLoggerの説明”にあります。GUI版とバッチ版の精度は、それぞれ64ビットと80ビットであることに注意してください。GUI版で精度が低くなるのは、JavaのJava Native Interfaceの機構に数値計算の精度の制約が存在するためです。

## 2 E-CELL2.25のデモンストレーション

それでは、具体的なモデルを用いて、E-CELL2を動かしてみましょう。この節では、ヒト赤血球の代謝モデルを用いて、E-CELL2を使ってみます。何故赤血球がシミュレーションの対象として適しているのでしょうか？ それには、以下のような理由が挙げられます。

- 成熟した赤血球細胞では、転写、翻訳、複製が行われず単純である。
- 実験材料として扱いやすく、大量の生化学的実験データの蓄積がある。

赤血球の主な代謝系は、解糖系、ペントースリン酸経路、核酸合成経路より構成されています。この他に、酸素運搬を行うヘモグロビンが大量に存在していて、赤血球は、これらの代謝系による自己調節機能を持った、大量のヘモグロビンが詰まった袋になぞらえることができます。今回のモデルでは、代謝経路の再構成を目的としていますが、将来的には、ヘモグロビンの運搬も別のモデルで表現され得ると考えられます。

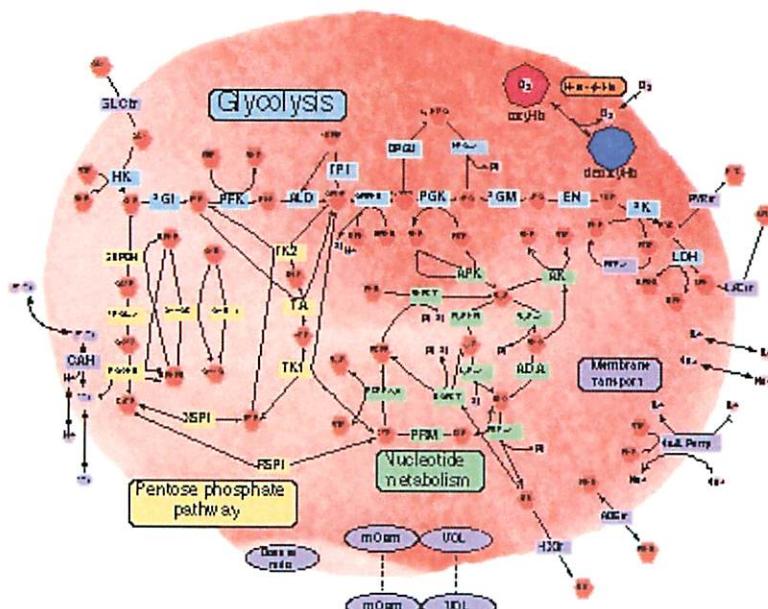


図4. ヒト赤血球の代謝経路の模式図

このヒト赤血球細胞の代謝モデルは、慶應義塾大学の中山講師、富田教授が中心になって構築されました。より詳細な情報については、the E-CELL.Org の研究紹介ページ(<http://www.e-cell.org/poster/ynakayam/ISMB99/ppframe.htm>)をご覧下さい。今回のデモンストレーションでは、正常なヒト赤血球細胞の代謝モデルを扱っていますが、遺伝子異常による貧血のシミュレーションも行われています。興味のある方は、中山講師([ynakayam@sfc.keio.ac.jp](mailto:ynakayam@sfc.keio.ac.jp))にメールを送って下さい。

赤血球細胞のシミュレーションを見るには、[スタート]から[プログラム]-[E-CELL2]-[erythrocyte]を選択するか、デスクトップ上のアイコンをクリックします。

赤血球モデル用のE-CELL2が起動します。E-CELL2は、“default.ecs”というファイルがECELL2.BATと同じディレクトリにあると、このファイルを読んでシミュレーション開始を試みます。今回の場合、図4の赤血球モデル用に default.ecsが用意されているので、自動的にデモンストレーションが開始されます。シミュレーションのための初期値や、パラメータを記述しているファイルは、“Erythrocyte\_v236.eri”です。

“.ecs”で表されるファイルは一般に、E-CELL2を自動化するためのスクリプトで、この中にE-CELL2を操作するための命令が書き込まれています。詳細は、2.3節と3.4節を参照して下さい。スクリプト実行により、図4に示した赤血球モデルのシミュレーションが開始されます。このスクリプトでは、500秒に達した時点で計算が停止します。

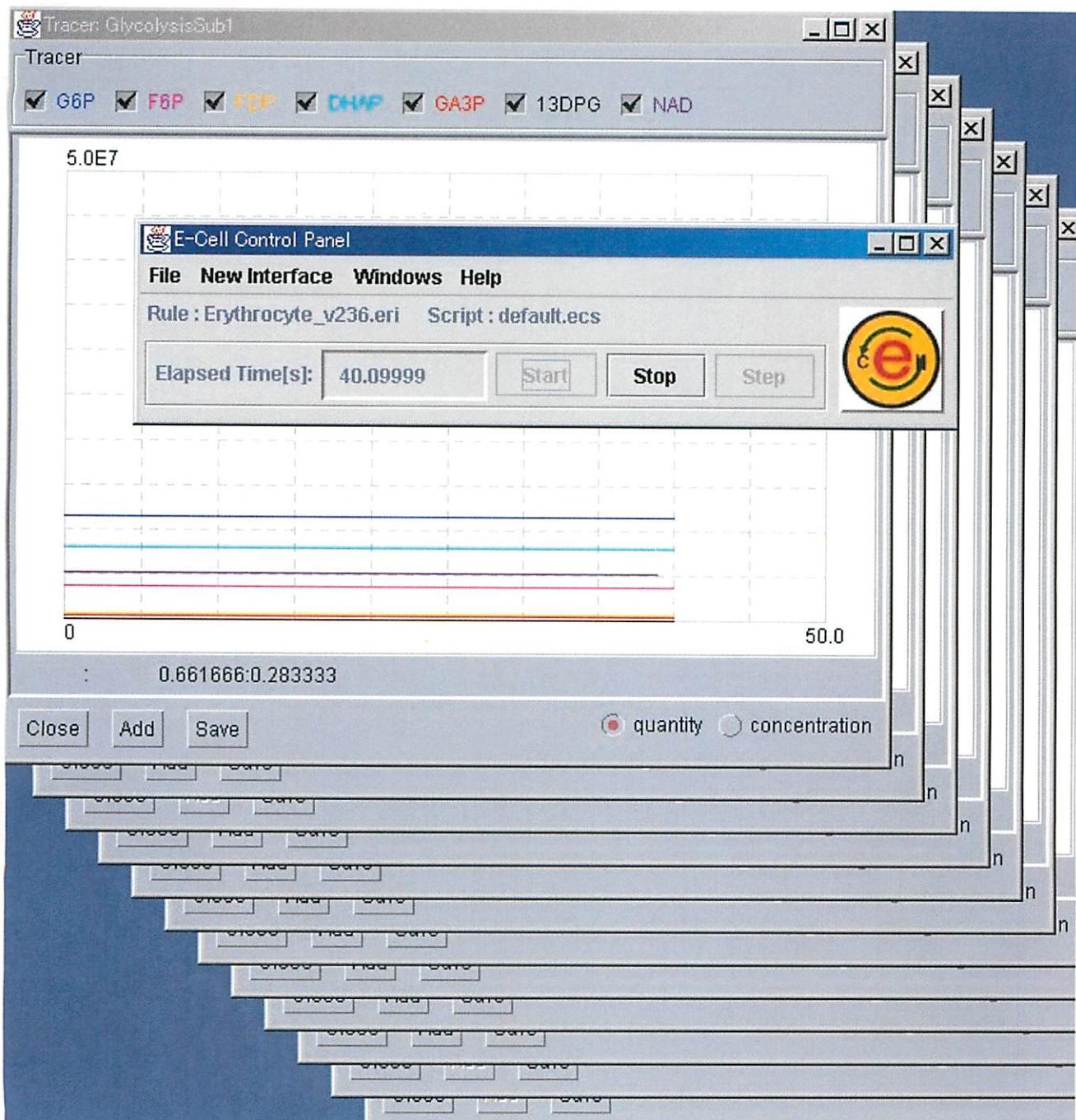


図5. 赤血球モデルのシミュレーション

スクリプトによる停止命令前にE-CELL2を終了させるには、コントロール パネルの“Stop”ボタンを押してシミュレーションを停止し、[File]メニューから[Quit]を選択します。

次の節では、E-CELL2の操作方法について、一通り解説されます。操作方法を理解したら、E-CELL2を再び起動して、色々と操作して遊んでみましょう。

### 3 E-CELL2のルール作成

具体的な説明に入る前に、シミュレーション用の簡単なモデルを紹介しておきます。下の図は、E-CELLモデルリスト達に“Toy(おもちゃ)”と呼ばれる、最も簡単なフィードバック系です。

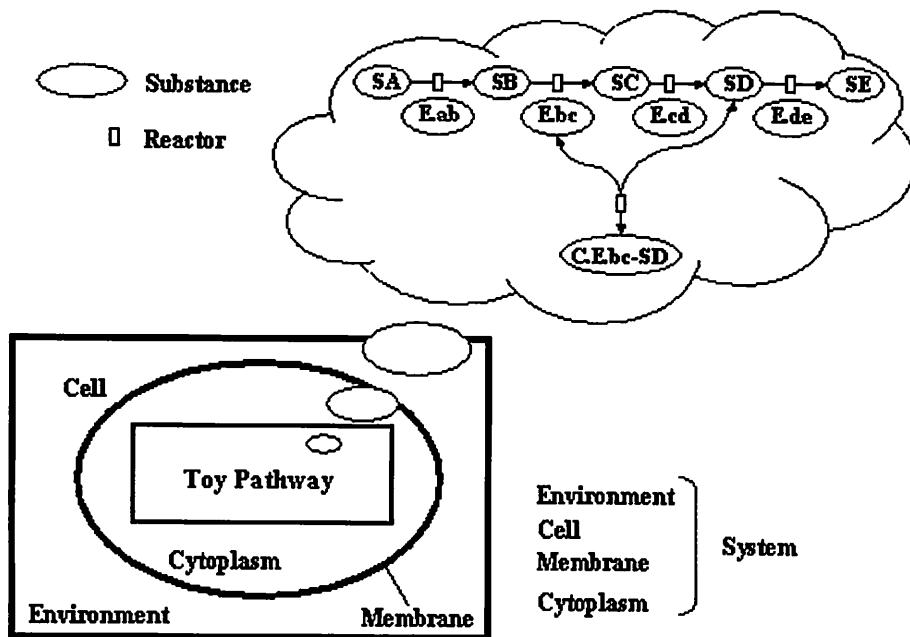


図6. Toyモデル

このモデルでは、5つの“S”ubstrate(基質)、4つの“E”nzyme(酵素)、1つの“C”omplex(複合体)が登場し、4つのReactor(化学反応式)によりSubstrate間が結びつけられます。Reactorは、例えばMichaelis-Menten反応のような化学反応式を体現しています。これらの反応は、培養細胞であれば培地に相当するEnvironment(環境)の中に浮かぶCell(細胞)の中に存在しています。細胞は、Membrane(膜)により外界と内部が区別され、細胞の内部はCytoplasm(細胞質)で満たされています。

これで、化学反応のネットワークを静的なグラフとして表現できました。

しかし、実際にシミュレーションを始めるためには、個々の反応が、どのようなパラメータ(例えば初期値や反応速度)のもとで、どのように反応が進むか(どの化学反応式を選ぶか)ルールを決めてあげる必要があります。

E-CELL用のモデルを作る時には、通常”.er”的拡張子で識別される、ルールファイルというシミュレーションのルールをテキストエディターで記述します。それから、E-CELLに付属するツールを使って、ルールファイルから、ルール・インター・メディエイト・ファイルという、E-CELLが読み込み可能なファイルを作成します。ルール・インター・メディエイト・ファイルは、通常”.eri”的拡張子で識別されます。

E-CELLを初めて使う人が、最初からルールファイルを記述するのは難しいので、ルールファイル作成を支援するために、スプレッド・シートから、ルールファイルを作成するツールも提供されています。

### 3.1 スプレッドシートの作成

ここではシミュレーション用スプレッド・シートを作成し、E-CELL2のルール・インター・メディエイト・ファイルに変換する方法の概略を説明します。

E-CELL2付属のモデリングランチャを使って、ルールファイルを作成するのに最低限必要な操作方法について説明します。

モデリングランチャを起動するには、[スタート]から[プログラム]-[E-CELL2]-[ModelingLauncher]を選択するか、デスクトップ上のアイコンをクリックします。

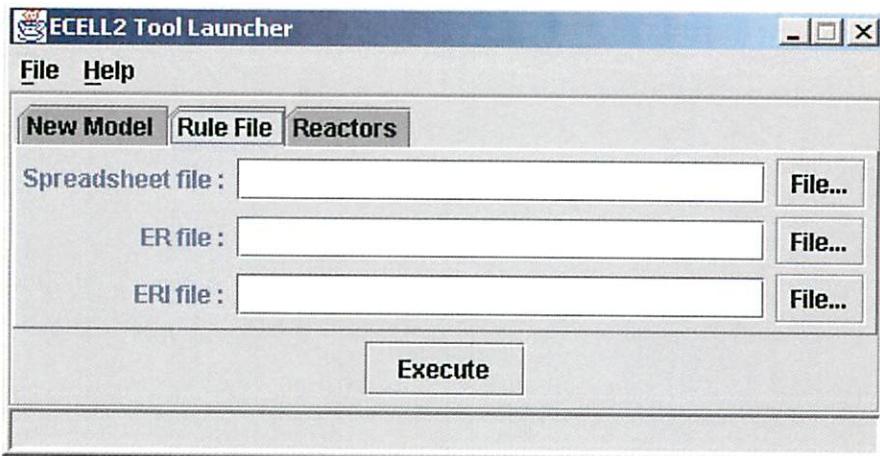


図7. モデリングランチャの画面(1)

E-CELL2のサンプルのルールファイルを読み込んでみましょう。Spread sheet fileの欄のFile..ボタンをクリックし、Editを選択しファイル選択画面を開きます。“sample.txt”を探し、このファイルを読み込んでください。

C:\E-CELL2\standard\sample.txt (Spreadsheet)								
0	1	2	3	4	5	6	7	
Type	Class	path	ID	Name	Inside	Outside	VolumeIndex	Memo
System	Cell	/	CELL	The cell				Def i
System	Environment	/	ENVIRONMENT	The cultu...			/ENVIRONM...	Def i
System	Cytoplasm	/CELL	CYTOPLASM	The cytop...			/CELL/CYT...	Def i
System	Membrane	/CELL	MEMBRANE	The membrane	/CELL:CYT... /:ENVIRON...			Def i
Type	Class	path	ID	Name	Arg_tag	Arg_coeff	init_act	Memo
Reactor	ConstantP...	/ENVIRONMENT	VOLUME	Volume in...	Value	1E-015	1E-015	Def i
Reactor	ConstantP...	/CELL/CYT...	VOLUME	Volume in...	Value	1E-018	1E-018	Def i
Type		path	ID	Name	QTY	CONC		Memo
Substance		/CELL/CYT...	SA	Substance A	1000000		1e+06 mee...	
					Fix		SA is fixed	
Substance		/CELL/CYT...	SB	Substance B	0			
Substance		/CELL/CYT...	SC	Substance C	0			
Substance		/CELL/CYT...	SD	Substance D	0			
Substance		/CELL/CYT...	SE	Substance E	0			
Substance		/CELL/CYT...	E.ab	Isomerase...		0.83027009		
					Fix			
Type		path	ID	Name	CONC	Arg_tag	Arg_coeff	
Substance		/CELL/CYT...	E.bc	Dehydrata...	0.02	Accumulator	SimpleAcc...	
Substance		/CELL/CYT...	E.cd	Isomerase...	0.01			
Substance		/CELL/CYT...	E.de	Isomerase...	0.01			
Substance		/CELL/CYT...	C.Ebc-D	Complex o...	0			
Type	Class	path	ID	Name	S ID	S path	S Coeff	P ID

図8. モデリングランチャの画面(2)

編集が終わったら、[File]メニューから、[Save]または[Save As]を選んで、保存してください。

### 3.2 ルールファイルとルール・インターメディエイト・ファイル作成

スプレッドシートからルールファイル、ルールファイルから ルール・インターメディエイト・ファイルを作成するための手順を説明します。ルールファイルについての詳細は、4章を参照してください。

モデリングランチャーを起動して、“sample.txt”と“toy.txt”を読み込みましょう。“toy.txt”は、Toyモデルのスプレッド・シートが一部未完成なファイルです。これらのファイルは、チュートリアル用に“standard”ディレクトリに置いてあります。

E-CELLのSubstance-Reactorモデルには、“System”、“Substance”、“Reactor”があると既に説明しました。スプレッドシートの一番左のカラムは、Typeと呼ばれ、各フィールドにはこれらの内のひとつが入ります。Systemは、細胞の構造や物質の局在を示すために使用され、代表的なものとして、“Environment(環境)”、“Cell(細胞)”、“Membrane(膜)”、“Cytoplasm(細胞質)”があります。Systemについての詳細な説明は、4.2.2節を参照して下さい。ここでは、“Inside(内部)”、“Outside(外部)”を区別することで、細胞等の入れ子構造が表現できることを覚えておいて下さい。

Substanceは広い意味での物質を意味します。スプレッドシートの中には、次の表のような記述があると思います。“Toy”モデルにおけるSubstanceは、化学反応のための材料と、反応を触媒する酵素になります。

表2.1 化学反応の材料となるSubstanceの記述例

Type	path	ID	Name	QTY
Substance	/CELL/CYTOPLASM	SA	Substance A	1000
				FIX
Substance	/CELL/CYTOPLASM	SB	Substance B	0
Substance	/CELL/CYTOPLASM	SC	Substance C	0

PathはSubstanceが存在する場所で、UNIXのディレクトリ・パス風に記述されます。IDは指定したSubstanceのIDで、必須項目です。Nameは指定したSubstanceの名称です。QtyとConcはどちらか片方を記述しなければならない項目で、QtyはSubstanceの“Quantity(個数)”, Concは“Concentration(濃度)”の初期値です。Concの単位にはmol/lを使用します。QuantityやConcをシミュレーション中に変化させない場合には、直下の行に“Fix”タグをつけて、数値を固定できます。Substanceの詳細については、4.2.3節を参照して下さい。

以下は酵素に関する記述です。

表2.2 酵素として反応を触媒するSubstanceの記述例

Type	path	ID	Name	CONC
Substance	/CELL/CYTOPLASM	E.bc	Enzyme B	0.02
Substance	/CELL/CYTOPLASM	E.cd	Enzyme C	0.01
Substance	/CELL/CYTOPLASM	C.E.bc-SD	Complex of E.bc and SD	0

### 〈課題〉

-  1. “toy.txt”中で、Substance DとSubstance Eの記述が一部抜けています。

す。”sample.txt”を見ながら埋めてみましょう。

- ~~※~~ 2. 同様に、Isomerase of AとIsomerase of Dの記述を補って下さい。

一方、Reactorは反応が実体化したもので、例えば以下のように定義されます。

表2.3 Reactorの記述例

Type	Class	path	ID	Name	S_ID	S_path
Reactor	MichaelisUniUniReactor	/CELL/CYTOPLASM	E.ab-0	A->B	SA	/CELL/CYTOF
Reactor	MichaelisUniUniReactor	/CELL/CYTOPLASM	E.bc-0	B->C	SB	/CELL/CYTOF
Reactor	MichaelisUniUniReactor	/CELL/CYTOPLASM	E.cd-0	C->D	SC	/CELL/CYTOF

PathはReactorが存在する場所で、UNIXのディレクトリ・パス風に記述されます。S\_IDは”Substrate(基質)”で、化学反応の出発物質を意味し、S\_Pathはその存在する場所です。P\_IDは”Product(産物)”で、化学反応の結果生成した物質を意味し、P\_Pathはその存在する場所です。C\_IDは”Catalyst(触媒)”で、反応を促進あるいは抑制する化学物質を意味し、C\_Pathはその存在する場所です。Arg\_tag列には、Reactorの種類によって定義されている、定数名が入力されます。Arg\_coeffには、Arg\_tagに対応する定数値が入力されます。

IDが!で始まるReactorは、裏口リアクターで、微分方程式では記述できない反応を記述するのに使用されます。“Toy”モデルでは、“!EQ-Ebc-D”が代数式の裏口リアクターです。 $Ebc+SD \rightleftharpoons C.Ebc-D$ という反応が、EbcとSDからC.Ebc-Dが生成する反応と、C.Ebc-Dの分解する反応の可逆反応なため、微分方程式では解くことができません。そのため、迅速平衡用の”RapidEquilibriumPReactor”を利用します。

#### 〈課題〉

- ~~※~~ 1. ReactorのD->Eの反応について、記述を補ってください。

Reactorの詳細については、4章2.4節”スプレッドシートの記述・Reactorパート”を参照して下さい。ここまで作業が終わったら、“toy.txt”を保存して、ルールファイルを作成する作業に入りましょう。

それでは、スプレッド・シートからルールファイルを作成する実際の操作をモデリングランチャーを使いながら説明していきましょう。

Spread sheet fileの欄の”File...”ボタンをクリックし、“Choose...”を選択しファイル選択画面を開きます。“toy.txt”を探し、このファイルを読み込んでください。

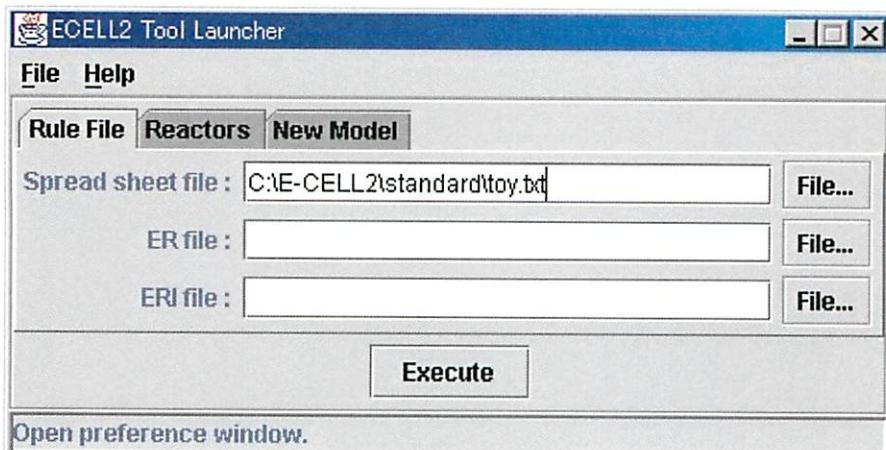


図9. toy.txtの選択

Executeボタンをクリックすると、ルールファイル“toy.er”とルール・インターメディエイト・ファイル“toy.eri”ファイルが作成されます。

ルールファイルやルール・インターメディエイト・ファイル作成に関する説明は以上です。次のReactorを作成するステップに進みましょう。

## 4 E-CELL2のユーザー定義Reactorの作成

標準リアクターにないような化学反応を記述したい場合は、あなた自身が、E-CELL2用にReactorをデザインする必要があります。

### 4.1 Reactorの概略について

- ReactorとはSubstanceの量の時間的変化を計算するものである。
- そのソースはC++言語を用いて記述している。
- E-CELLでReactorを書くとはリアクター定義(Reactor Description, RD)ファイル(*filename.rd*)を書くことである。
- RDファイルには、Reactorの仕様と実行される処理の内容が書かれている。
- rd2ch.pl, rd2tex.plを使ってRDファイルはtexファイル(LaTeX形式のReactor Spec Sheet)や”.dll”ファイル(E-CELL Systemがロードできる形式)などに変換される。

RDファイルとはキーワードと値の組からなる行を、必要な数だけ並べて作成するものです。

RDファイルの構成は大きく

- 一般情報に関する部分
- Reactor Spec Sheetに関する部分
- Reactor Source Codeに関する部分

の3つに大別できます。

リアクター定義ファイルを作成する際には以下の約束事に注意する必要があります。

- キーワードはアルファベットの大文字又は「\_」(下線記号)で構成され、スペースを含まない。
- キーワードは「@」または「%」で始まる。「@」で始まる行は単純にその内容が読み込まれる。  
「%」で始まる行はその中身を「,」(カンマ記号)で区切ることによって、配列として処理される。
- #で始まる行はコメント文とみなされる。行頭の #をそのまま出力する際は、##とするかス

- ベースを1つ入れる。ただし、行頭以外の # はそのまま出力される。
- 行頭にキーワードのない行は、それより前の行の内容に続くものとして解釈される。(Spec Sheet に変換する際に改行したい場合、改行したい箇所に##を入れる。)
- キーワードは必要でないものに関しては省略できる。

RDファイルの例を以下に示します。

1. これはMichaelisUniUniReactorについてのファイルです。このReactorは次の反応速度式に従います。

$$v = \frac{K_c F [E][S]}{K_m S + [S]}$$

MichaelisUniUniReactorについてのファイル

```
@CLASSNAME:MichaelisUniUniReactor
```

```
@BASECLASS: FluxReactor
```

```
@AUTHOR: E-CECLL Tutorial
```

```
@EMAIL: tutorial@e-cell.org
```

```
@DATE: 2000 12/12
```

```
%VERSION: ecs-v1, 0.1
```

```
@BRIEF_DESCRIPTION:Unireactor enzyme activity of which kinetics can be described by the Henri-Michaelis-Menten equation.
```

```
@DESCRIPTION:A reactor class for unireactant enzyme activity where kinetics can be described by the Henri-Michaelis-Menten equation derived from rapid equilibrium assumptions.
```

```
#vspace {0.2cm}
```

This reactor is applicable to the following reaction sequence:

```
#begin {center}
$E+S #rightleftharpoons ^{[k_{(1)}]_{(-1)}} ES
#rightarrow ^{[k_{(p)}]} E+PS
#end {center}
#vspace {0.3cm}
```

```

@EQUATION: $$v=\frac{(K_{\text{cF}} [E] [S])}{(K_{\text{mS}}+[S])} $$

%SUBSTANCE:Substrate, 1, 1
%SUBSTANCE:Product, 1, 1
%SUBSTANCE:Catalyst, 1, 1
%SUBSTANCE:Effector, 0, 0

```

```

%PARAMETER: KmS, Float, mol/l, Michaelis Constant of Substrate
%PARAMETER: KcF, Float, mol/l, Catalytic Constant (Forward)

```

@REACT\_FUNC:

```
Float S = substrate(0) -> concentration();
```

```
Float E = catalyst(0) -> quantity();
```

```
Float velocity = KcF * E * S;
```

```
Float Den = KmS + S;
```

```
velocity /= Den;
```

```
process(velocity);
```

## 2. 一般情報に関するキーワード

- @CLASSNAME: 作成するReactorのクラス名(ファイル名から.rdを削除した形)
- @BASECLASS: そのクラスが継承する元の基底クラス
- @AUTHOR: 作成者名
- @EMAIL: E-Mail address
- @DATE: 作成日
- %VERSION: E-CELLSystemのバージョン, このReactorのバージョン
- @BRIEF\_DESCRIPTION: そのReactorの簡単な説明

## 3. Reactor Spec Sheetに関するキーワード

- @DESCRIPTION: そのReactorの詳しい説明
- @EQUATION: Reactorの式を LaTeX の displaymath 環境で以下のいずれかの方法で記述
  1. \$\$<\text{数式の記述}>\$\$
  2. \$\begin{displaymath}<\text{数式の記述}>\end{displaymath}\$
  3. \$\left[<\text{数式の記述}>\right]\$
- %SUBSTANCE: Substance の定義で、Substrate、Product、Catalyst、Effector の 4 種類が用意されています
- @NOTES: Reactor の実装に関する注意点を書きます

#### 4. Reactor Source Codeに関するキーワード

- %PARAMETER: パラメータ名、パラメータの型、単位、パラメータに関する記述を、  
　　``,"で区切って書きます
- @PRIVATE: オブジェクトの私的要素(できる限りメンバ変数はこれで定義しましょう)
- @PROTECTED: オブジェクトの限定公開要素(派生クラスからは参照できます)
- @PUBLIC: オブジェクトの公開要素
- @INITIALIZE\_FUNC: ここでは主にパラメータの値域のチェックや、シミュレーション中に変化しない値の計算などの初期設定を行います
- @REACT\_FUNC: 每ステップ行う処理を書きます。1.反応速度を算出し、2.その速度にしたがって物質の量を増減させる、といった一連の処理を書きます。Process()メソッドの引数は、1秒あたりの反応分子数で、Float型です。

以上がReactorに関する簡単な説明です。Reactorの詳細については、5章を参照してください。

## 4.2 Reactorの作成手順

簡単な例を挙げて、Reactorをどのように作成していくかを説明します。

モデリングランチャーを起動し、Reactorsのタブをクリックします。RD fileの欄の“File..”ボタンをクリックし、“Edit...”を選択しファイル選択画面を開きます。

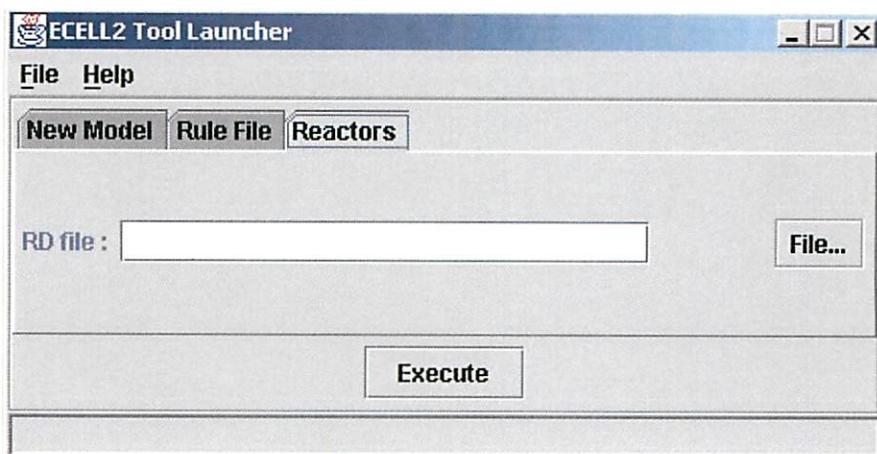


図10. Reactorタブ

“MyMichaelisMentenUniUni.rd”というファイルを開いてみましょう。リアクター定義 ファイル(RDファイル)は、通常”.rd”という拡張子を持つファイルとして作成されます。このファイルの中には、リアクターの定義が記述されています。

```

C:\E-CELL2\standard\RDL\MyMichaelisUniUniReactor.rdl (Text)

File
64CLASSNAME: MyMichaelisUniUniReactor

@BASECLASS: FluxReactor
@author: Yusuke Saito
@email: t96406ys@sfc.keio.ac.jp
@date: 29/6/1999

#@VERSION: E-CELL, Reactor
%VERSION: ecs-v09, 0.1

@BRIEF_DESCRIPTION: Unireactant enzyme activity of which kinetics can be described by the Henri-Michaelis-Menten equation

@DESCRIPTION: A reactor class for unireactant enzyme activity where kinetics can be described by the Henri-Michaelis-Menten equation derived from rapid equilibrium assumptions.
$vspace{0.2cm}

This reactor is applicable to the following reaction sequence:

$begin{center}
$E+S \rightleftharpoons ^{k_{\{1\}}}_{k_{\{-1\}}} ES
\rightarrow ^{k_{\{p\}}} E+P
$end{center}

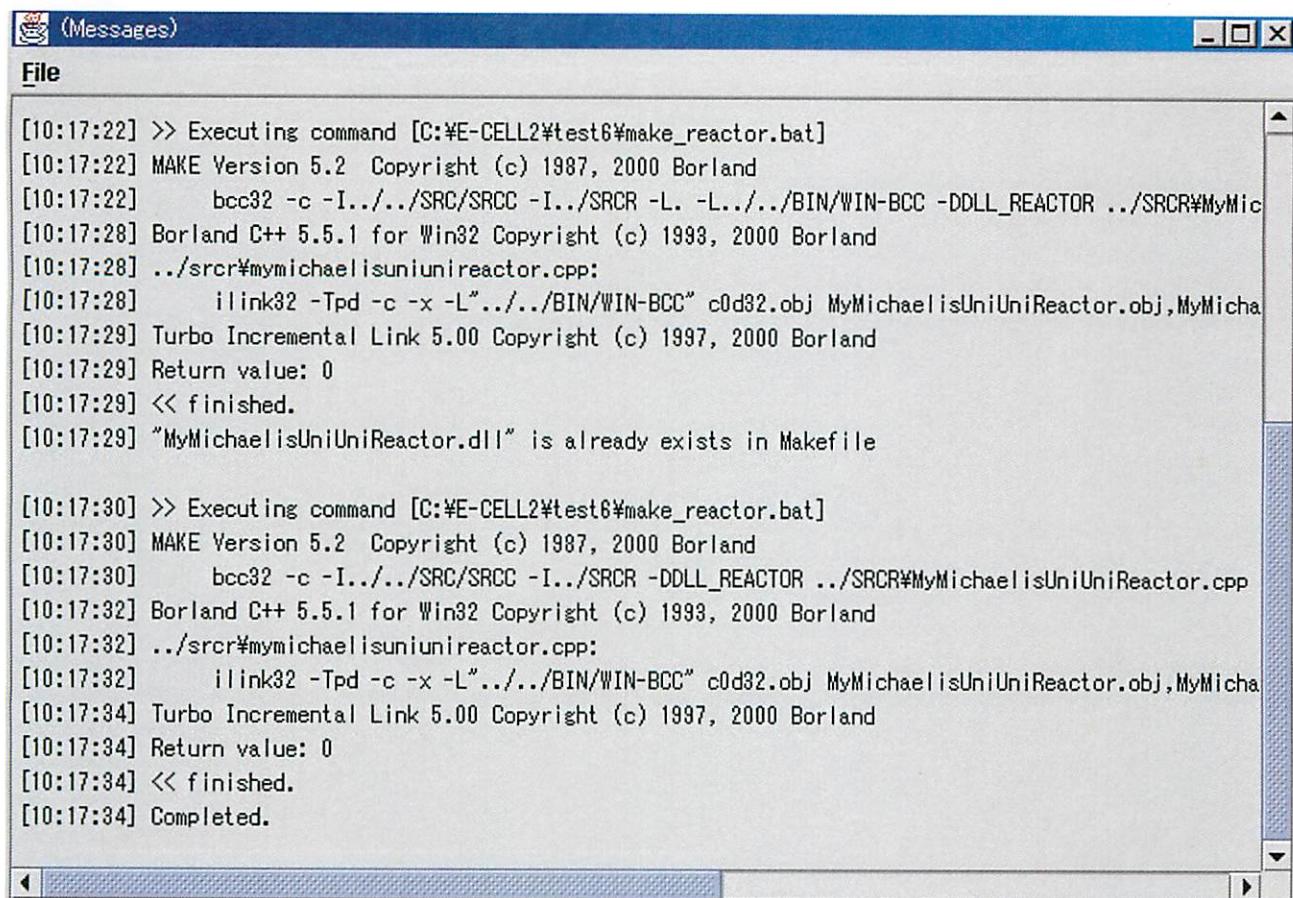
```

図11. リアクタ一定义ファイルの例

リアクタ一定义ファイルを作成する上でまず注意する必要があるのは、**拡張子を除いたファイル名が、そのまま@CLASSNAMEとならなければならぬ点です。**このファイルでも一致させてあります。MyMichaelisMentenUniUniReactor.rdlは、クラス名を除いては MichaelisMentenUniUniReactor.rdlと内容が同一です。標準リアクターの定義ファイルは、標準的なインストールでは、“C:\E-CELL2\standard\RDL”にあります。実際にリアクタ一定义ファイルを作成する際は、これらのファイルの記述を参考にすると良いでしょう。

リアクタ一定义ファイルから、モデリングランチャーを利用してC++のソースコードとヘッダファイルを生成し、コンパイルを行なうことでリアクタ一DLLを作成します。“Execute”ボタンを押してDLLを作成しましょう

RDファイル中のC++コードに文法的な誤りがなく、Borland C++ Compilerの設定が適切に行われていれば、DLLが作成されます。実行後にはメッセージウィンドウにコンパイル時のメッセージが表示されます。



```
[10:17:22] >> Executing command [C:\E-CELL2\test6\make_reactor.bat]
[10:17:22] MAKE Version 5.2 Copyright (c) 1987, 2000 Borland
[10:17:22]     bcc32 -c -I../../SRC/SRCC -I../../SRCR -L -L../../BIN/WIN-BCC -DDLL_REACTOR ../../SRCR\MyMic
[10:17:28] Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland
[10:17:28] ../../srcr\mymichaelisuniunireactor.cpp:
[10:17:28]     ilink32 -Tpd -c -x -L"../../BIN/WIN-BCC" c0d32.obj MyMichaelisUniUniReactor.obj,MyMicha
[10:17:29] Turbo Incremental Link 5.00 Copyright (c) 1997, 2000 Borland
[10:17:29] Return value: 0
[10:17:29] << finished.
[10:17:29] "MyMichaelisUniUniReactor.dll" is already exists in Makefile

[10:17:30] >> Executing command [C:\E-CELL2\test6\make_reactor.bat]
[10:17:30] MAKE Version 5.2 Copyright (c) 1987, 2000 Borland
[10:17:30]     bcc32 -c -I../../SRC/SRCC -I../../SRCR -DDLL_REACTOR ../../SRCR\MyMichaelisUniUniReactor.cpp
[10:17:32] Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland
[10:17:32] ../../srcr\mymichaelisuniunireactor.cpp:
[10:17:32]     ilink32 -Tpd -c -x -L"../../BIN/WIN-BCC" c0d32.obj MyMichaelisUniUniReactor.obj,MyMicha
[10:17:34] Turbo Incremental Link 5.00 Copyright (c) 1997, 2000 Borland
[10:17:34] Return value: 0
[10:17:34] << finished.
[10:17:34] Completed.
```

図12. コンパイル時のメッセージ

コンパイルが終わったら、実際に“MyMichaelisMentenUniUniReactor”を 使用してみましょう。モーデリングランチャーを利用して“toy.txt”を開きます。

“MichaelisMentenUniUniReactor”と “MyMichaelisMentenUniUniReactor”と差し替えて、ファイルを保存 し、“Execute”ボタンを押しましょう。

C:\E-CELL2\standard\toy.txt (Spreadsheet)					
File					
0	1	2	3	4	5
Type		path	ID	Name	CONC
Substance		/CELL/CYT...	E.bc	Dehydrat...	0.02
Substance		/CELL/CYT...	E.cd	Isomerase...	0.01
Substance			E.de	Isomerase...	
Substance		/CELL/CYT...	C.Ebc-D	Complex o...	0
Type	Class	path	ID	Name	S_ID
Reactor	MichaelisUniUniReversibleReactor	/CELL/CYT...	E.ab-0	Isomeriza...	SA
Reactor	MichaelisUniUniReversibleReactor	/CELL/CYT...	E.bc-0	Dehydrat...	SB
Reactor	MichaelisUniUniReversibleReactor	/CELL/CYT...	E.cd-0	Isomeriza...	SC
Reactor	MyMichaelisUniUniReactor	/CELL/CYT...	E.de-0	Isomeriza...	SD
Reactor	RapidEquilibriumPReactor	/CELL/CYT...	!EQ-Ebc-D	Bonding o...	E.bc
					SD

図13. ルールファイルの編集

ここで、E-CELL2を起動し、“toy.eri”を[File]メニューより [Load Rule]を選んで読み込み、シミュレーションを開始してみましょう。“sample.eri”と読んで実行したときと同じ挙動であれば、リアクター作成は成功です。

今後は、必要に応じてReactorをデザインして、独自のシミュレーションができるように訓練を積んで行ってください。

## 5.E-CELL2の操作方法

作成したルール・インターメディエイト・ファイルとリアクターを用いて、E-CELL2の基本的な操作を覚えましょう。E-CELL2を起動すると、まずコントロールパネルが現れます。

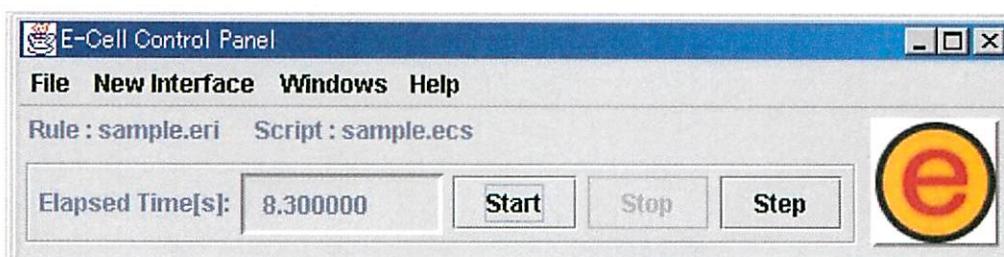


図14. E-CELL2のコントロールパネル

まず、プルダウンメニューについて簡単に説明します。

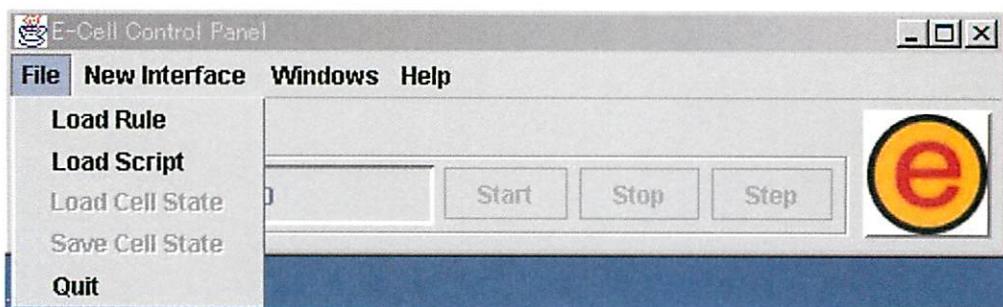


図15. [File]メニュー

[File]メニューには、[Load Rule]、[Load Script]、[Load Cell State]、[Save Cell State]、[Quit]の項目があります。

[Load Rule]は、ルール・インター・メディエイト・ファイルを読み込む際に選択します。

[Load Script]は、E-CELL2の操作を自動化するためのスクリプトを読み込む際に選択します。

[Load Cell State]は、途中経過を記録したファイルを読み込んで、シミュレーションを再開する際に選択します。

[Save Cell State]は、シミュレーションの途中経過を記録したファイルの書き出しに使用します。

[Quit]を選択すると、E-CELL2が終了します。

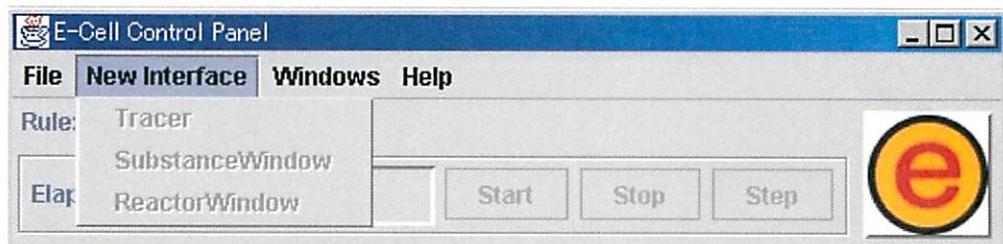


図16. [New Interface]メニュー

[New Interface]メニューには、[Tracer]、[SubstanceWindow]、[ReactorWindow]の項目があります。

[Tracer]は、Substanceの量や、Reactorの活性を描画する、[Tracer]画面を生成するために選択します。

[SubstanceWindow]は、Substanceの表示と操作を行う画面を開くために選択します。

[ReactorWindow]は、Reactorの表示を行う画面を開くために選択します。

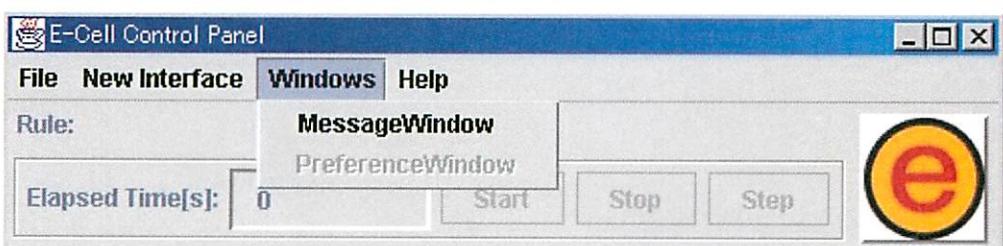


図17. [Windows]メニュー

[Windows]メニューには、[MessageWindow]と[PreferenceWindow]項目があります。

[MessageWindow]は、E-CELL2からのメッセージ画面を表示させるのに 使用します。

[PreferenceWindow]は、シミュレーションの1ステップの時間幅と、[Tracer]画面更新のタイミングを設定する際に選択します。

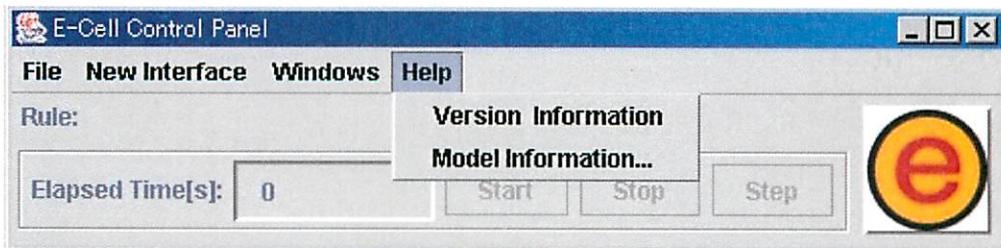


図18. [Help]メニュー

[Help]メニューには、[Version Information]項目があり、これを選択すると バージョン情報が表示されます。

以上が、メニューに関する説明です。

ルール・インター・メディエイト・ファイルを読み込むには、[File]メニュー -[Load Rule]を選択します。ファイルを選択する画面が表示されるので、ルール・インター・メディエイト・ファイルを意味する、".eri"で終わるファイルの名前をクリックし、Ok (開く)"を選択しましょう。

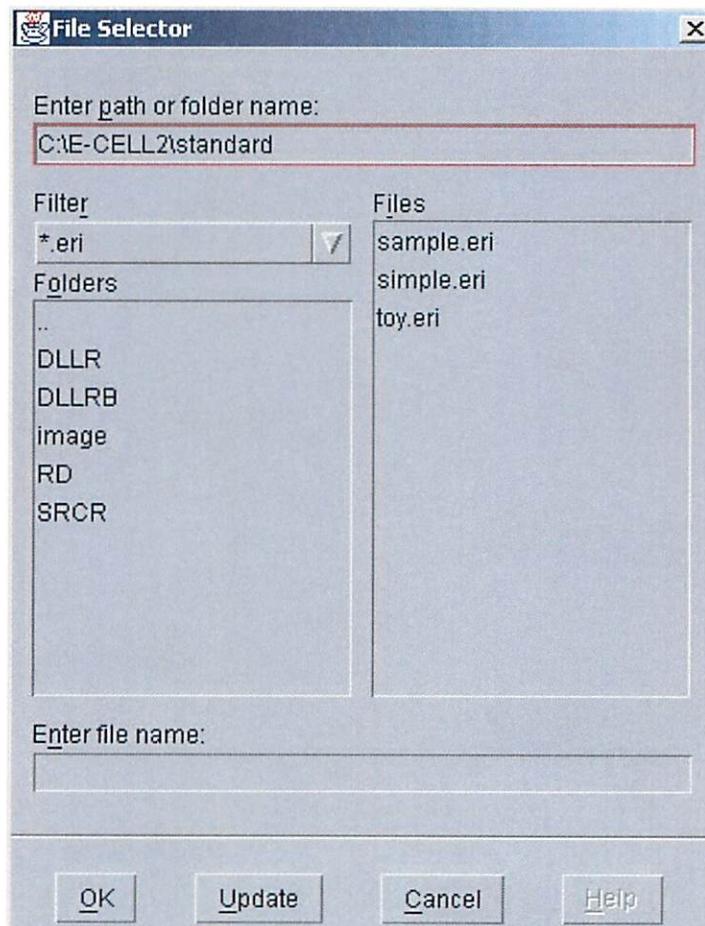


図19. ルール・インター・メディエイト・ファイル選択画面

指定されたファイルが正常に読み込まれると、Message Windowに、"Condition Good."と表示されます。

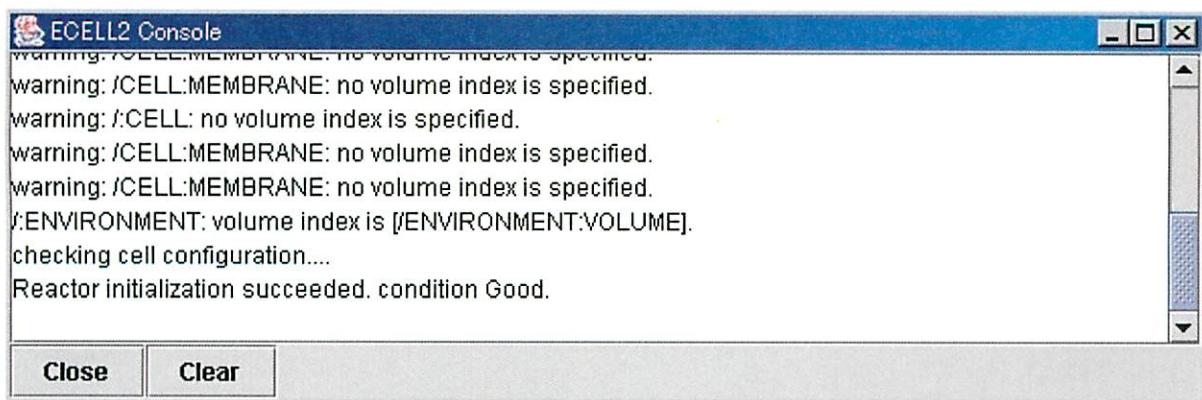


図20. ルール・インターメディエイト・ファイル読み込み時メッセージ

次に、[Tracer]画面を開いて、Substanceの量をグラフとして表示するための準備をします。[New Interface]-[Tracer]を選択するとTracerが表示されます。

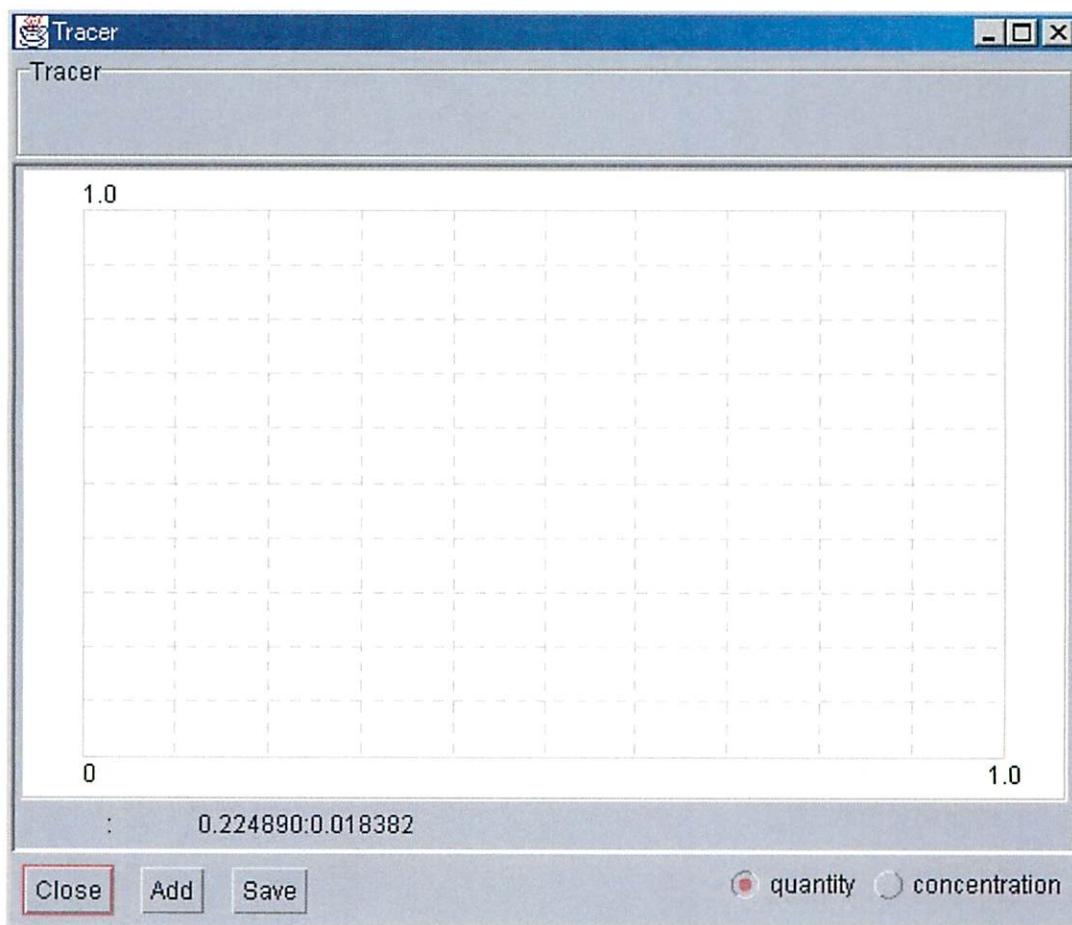


図21. Tracerの画面

Tracerの左下にある"Add"ボタンを押して、[Entry Selector]を表示します。左パネルをクリックして、Cytoplasmに位置しているSubstanceを選択してみましょう。

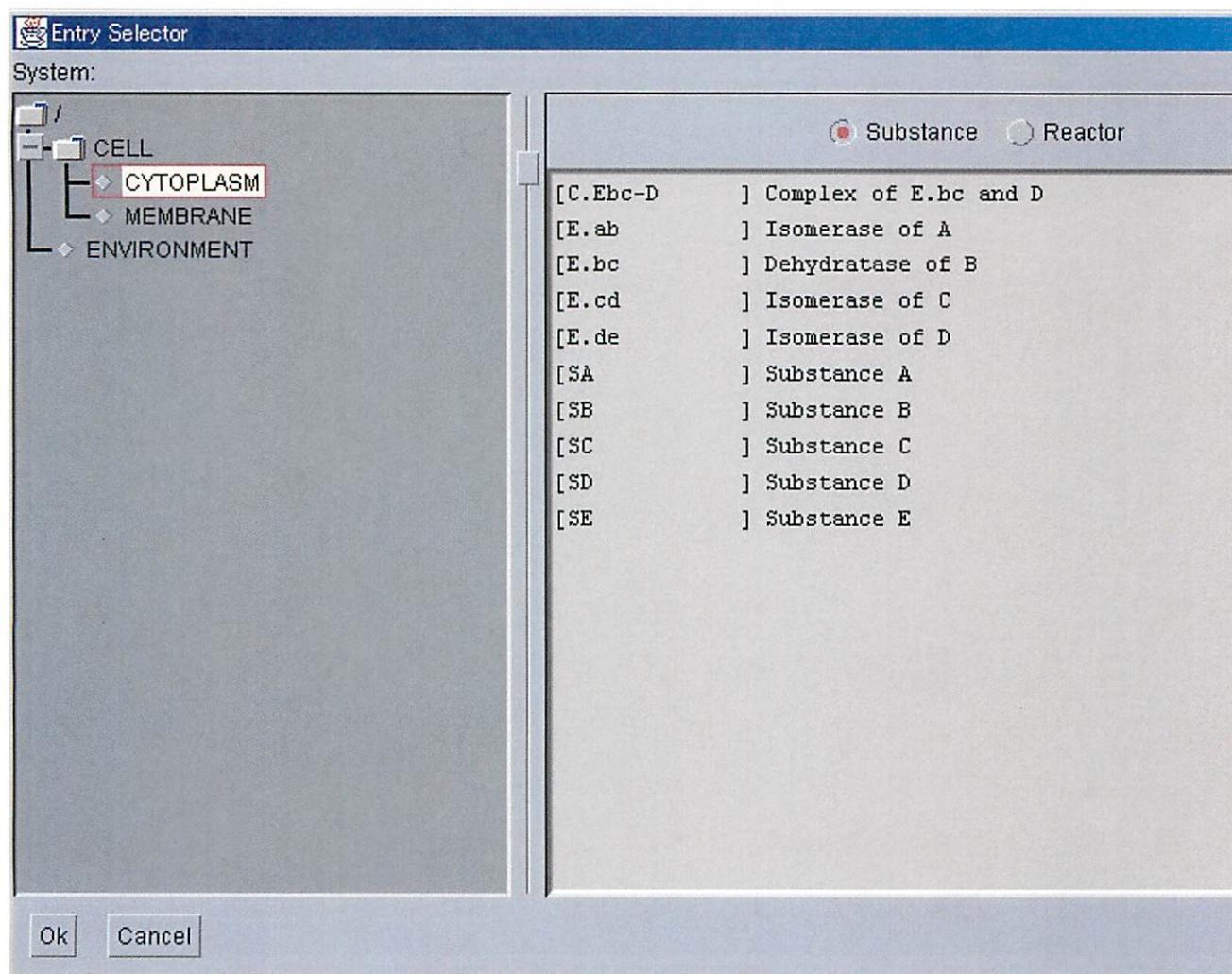


図22. Entry Selectorの画面(1)

Substanceのラジオボタンをクリックすると、Substanceが右のパネルに表示されます。Reactorを表示させることも可能です。

複数のSubstanceやReactorを選択する方法を説明します。まとめて選択するには、“Shift”キーを押しながら、マウスを左クリックしてください。飛び飛びに選択するには、“Ctrl”キーを押しながらマウスを左クリックしてください。選択したら、“Ok”ボタンを押します。この状態で、コントロールパネルの“Start”を押してシミュレーションを開始するようになります。

一枚のTracerには、8種類までのSubstanceやReactorを表示できます。

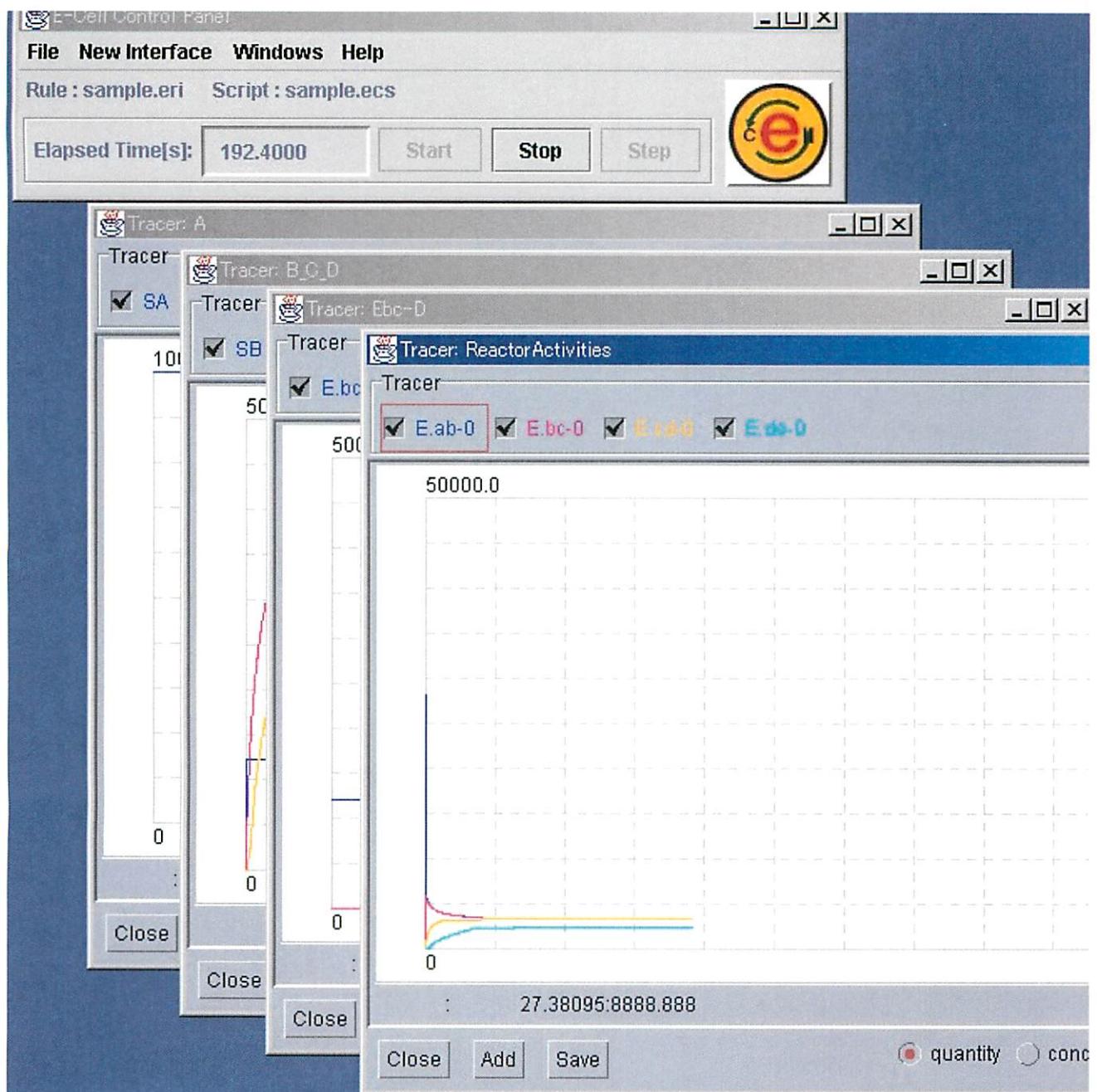


図23. E-CELL2のシミュレーションの様子

ルールに従って、Substanceの量や、Reactorの活性の変化がTracerに表示されます。Tracerに表示されたシミュレーションの結果は、Tracerの“Save”ボタンを押すと保存できます。詳細は、3.2.7節を参照して下さい。

Substanceの量を手動で変更して、全体の系にどのような影響が及ぶか見ることもできます。[New Interface]メニューから、[Substance Window]を選んでみましょう。

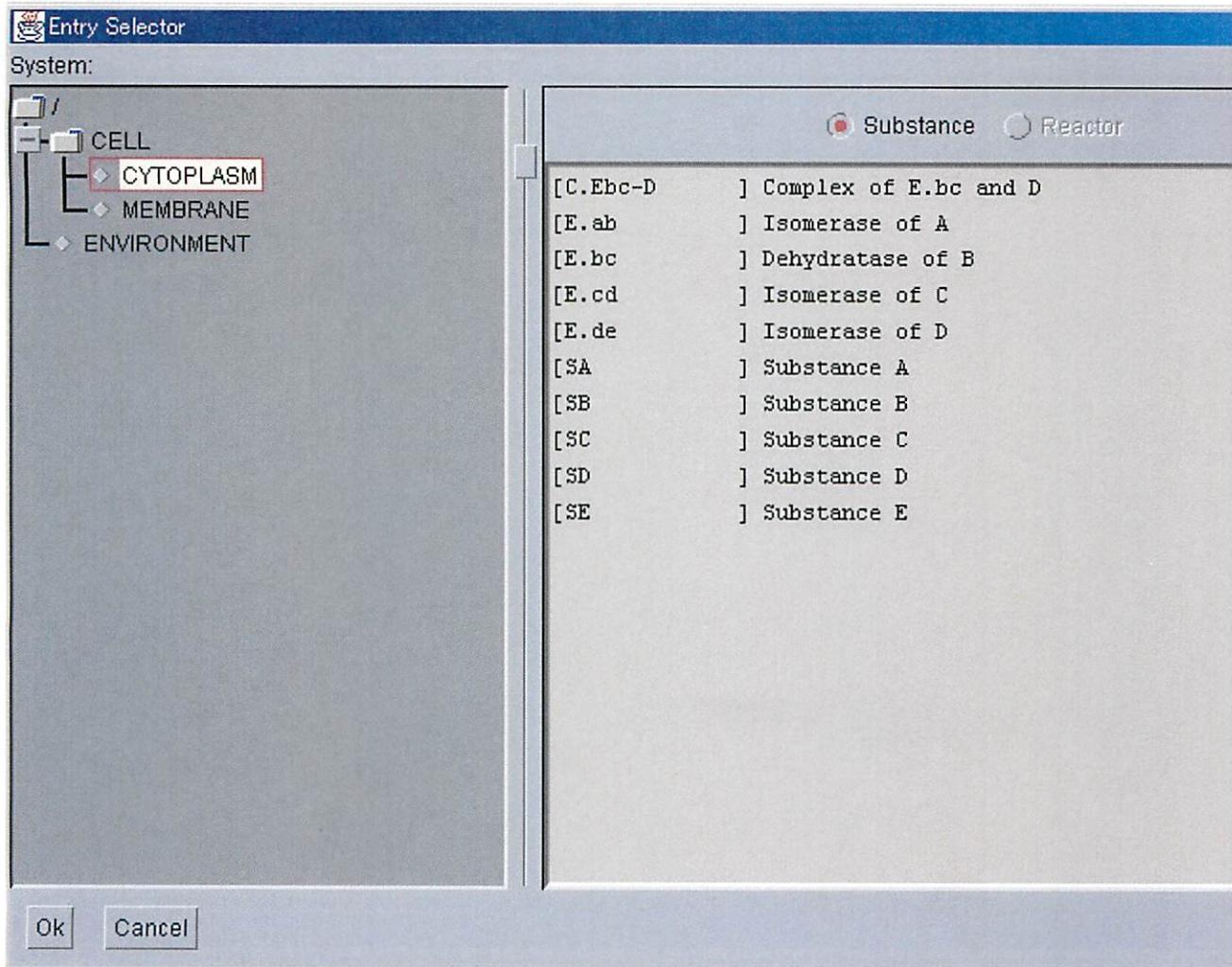


図24. SubstanceWindow用のEntry Selectorの画面

このEntry Selectorは、Substance専用なので、Reactorを選ぶことはできません。Substanceを選択し、“Ok”ボタンを押すと、選択されたSubstanceの個々の詳細情報を表示する画面が現れます。

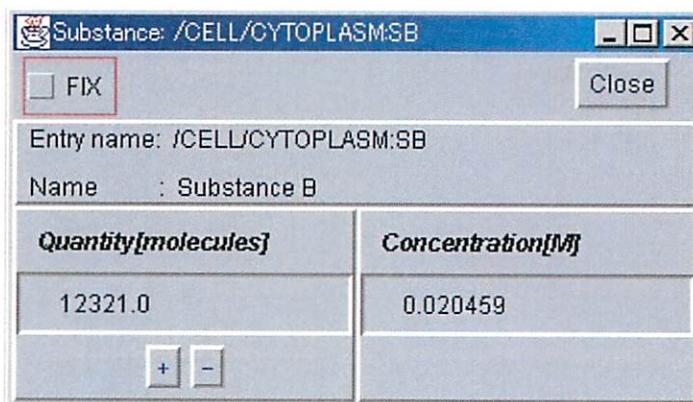


図25. SubstanceWindowの画面

“+”, “-”ボタンを押すか、“Quantity”あるいは“Concentration”テキストフィールドに値を直接入力して、注目するSubstanceの量を人為的に増減することができます。“Fix”チェックボタンをチェックすると、そのSubstanceの量を、計算結果を無視して一定に保つことが可能になります。 SubstanceWindow の操作を終了するには、“Close”ボタンを押します。

Substanceの量を変化させて、系にどのような影響が出るか見てみましょう。

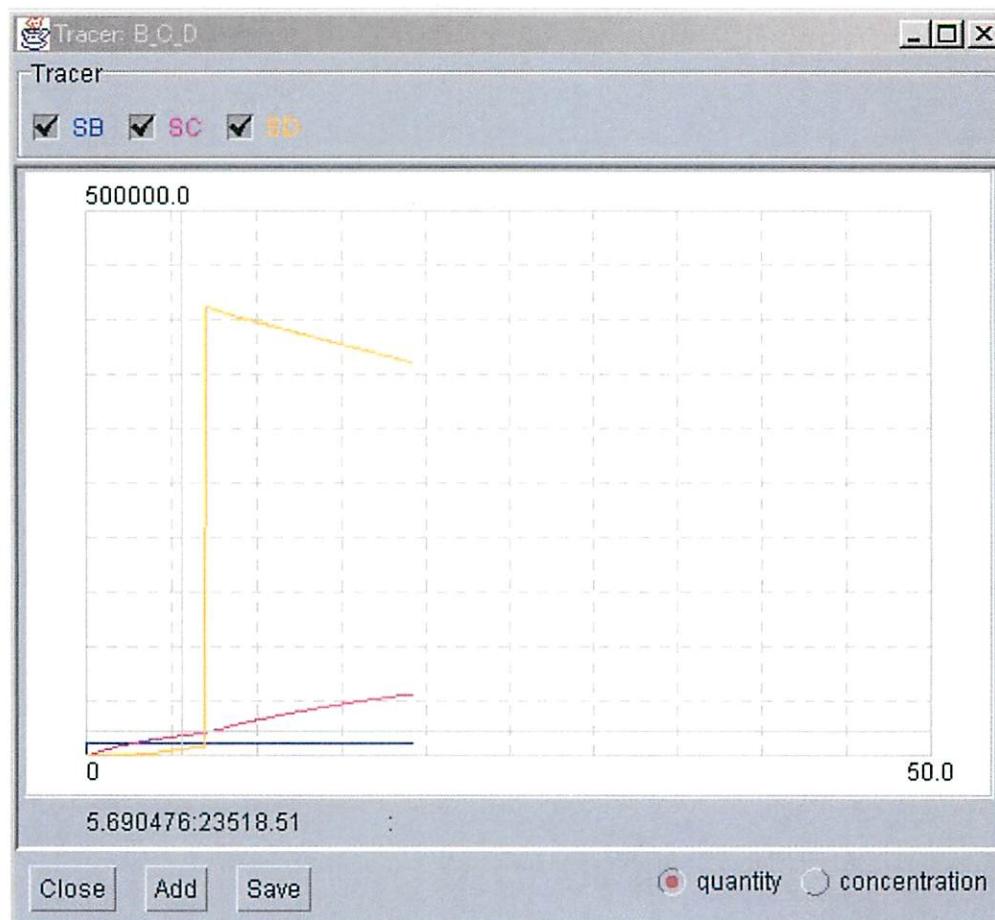


図26. Substanceの量の変更により生じた、系の振る舞いの変化の例

この例では、“Substance D”を急激に増加させています。このSubstanceを 増加させても、系の中の反応に補償されて、徐々に量が減っていくのが 分かります。

同様に、Reactorの活性を詳細表示することもできます。コントロールパネルの、[New Interface]メニューから、[ReactorWindow] を選びましょう。Reactor用の[Entry Selector]が表示されます。

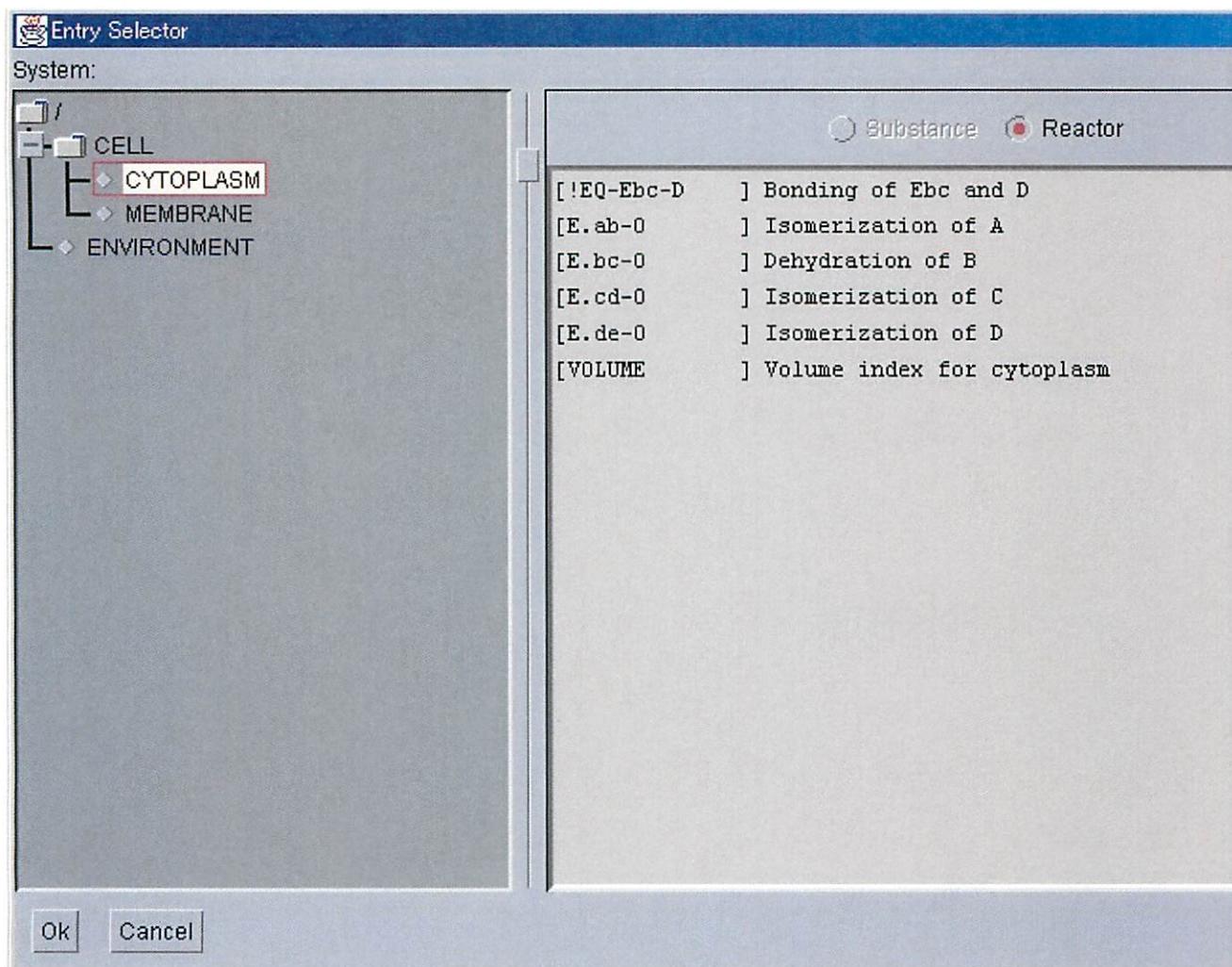


図27. ReactorWindow用のEntry Selectorの画面

SubstanceWindowのときと同様に、左パネルに表示されるSystemをクリックして、目的のReactorを探します。Reactorを選択したら、“Ok”ボタンをクリックしましょう。

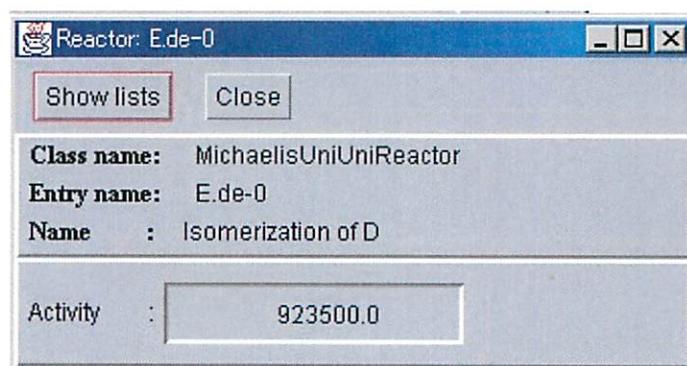


図28. ReactorWindowの画面

この画面では、あるReactorについて、その名前、使用している反応式の種類、反応の活性値が表示されます。“Show lists”ボタンを押すと、更に詳細な情報が表示され、“Close”ボタンを押すと画面が閉じます。

“Show lists”ボタンを押してみましょう。

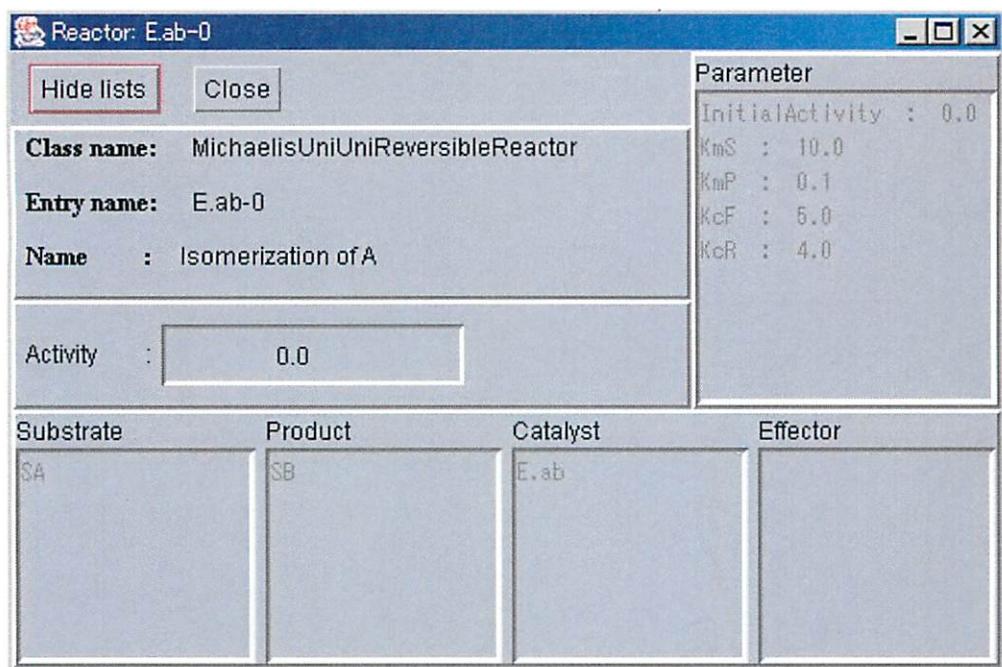


図29. ReactorWindow拡大画面

上記の情報の他に、各種パラメータの初期値、反応に関与する物質についての情報が表示されます。画面を元の大きさに戻す場合は、“Hide lists”ボタンを押し、閉じるには“Close”ボタンを押します。

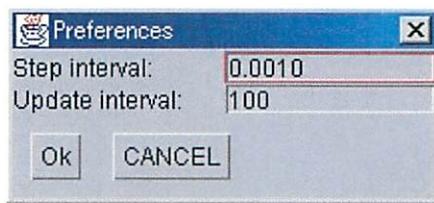


図30. PreferenceWindowの画面

次は、PreferenceWindowです。この画面を呼び出すには、コントロール パネルの[Windows]メニュー一から、[PreferenceWindow]を選びます。この画面は、シミュレーションのための計算の1ステップの時間幅(単位: 秒)と、Tracerの画面更新を何ステップ毎に行うかを制御するために用意されています。この時間幅を短く取ると、計算の精度は上がりますが、シミュレーションにかかる時間は長くなります。逆に、時間幅を長く取ると、シミュレーションにかかる時間は短くなりますが、計算の精度が荒くなり、結果を信頼することができなくなってしまいます。初期設定では、計算の1ステップが0.001秒、画面の更新頻度が100回に1回に設定されています。

この画面はモーダルダイアログなので、“Ok”ボタンで決定するか“CANCEL”で閉じるかしないと、他の画面の操作を行うことができません。

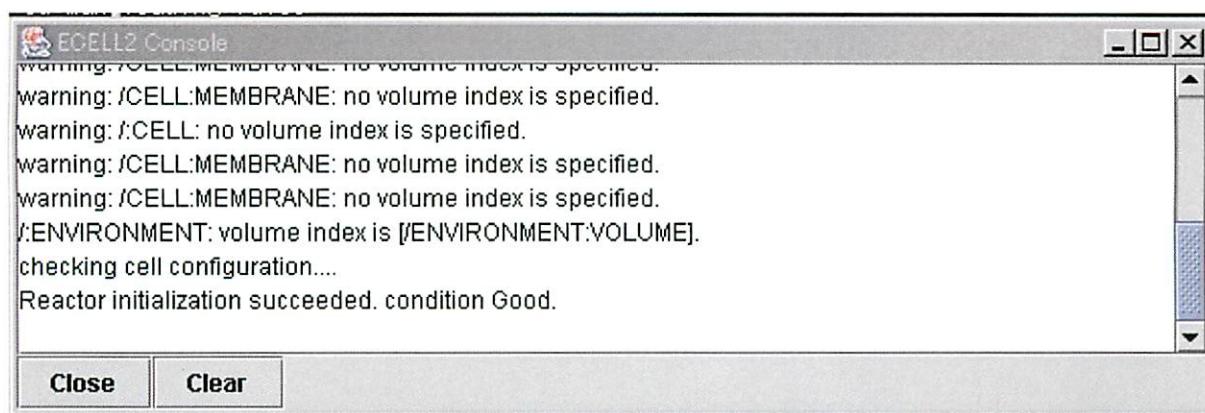


図31. MessageWindowの画面

MessageWindowには、E-CELL2からのメッセージが表示されます。このウィンドウを閉じるには、“Close”ボタンを押して下さい。“Clear”ボタンを押すと画面上に表示されたメッセージがクリアされます。

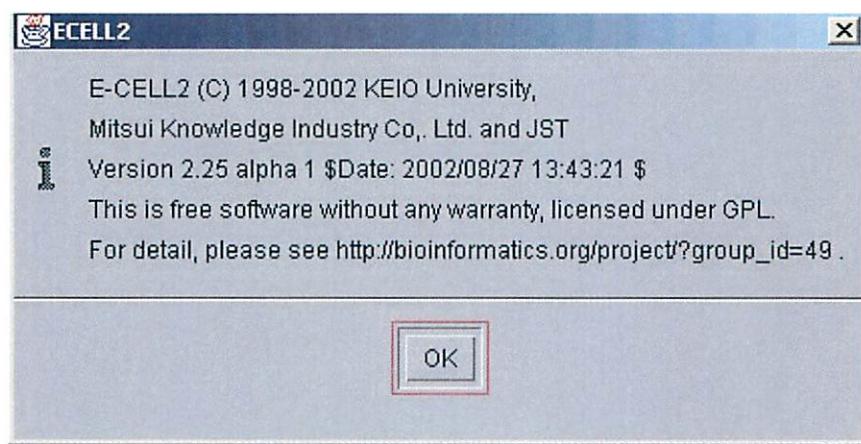


図32. Version Informationの画面

コントロールパネルの[Help]メニューから、[Version Information]を選択することができます。Version Information には、E-CELL2のバージョンの情報が表示されます。

Toy model - Microsoft Internet Explorer

File Edit View Favorites Tools Help

## Toy model (sample)

A sample "Toy" model that includes a simple feedback regulation mechanism, as illustrated in the figure below.

The diagram illustrates the Toy model. At the top, a legend indicates that an oval represents a Substance and a rectangle with a double-headed arrow represents a Reactor. Below this, a large cloud-like shape represents the Environment. Inside the environment, five substances (SA, SB, SC, SD, SE) are connected by reactors in a linear sequence: SA → SB → SC → SD → SE. Each substance is associated with an enzyme: E<sub>ab</sub>, E<sub>bc</sub>, E<sub>cd</sub>, and E<sub>de</sub>. A feedback loop is shown where the product SE activates the enzyme E<sub>bc</sub>. Below the environment, a rectangular box labeled "Cell" contains a smaller oval labeled "Toy Pathway". This pathway is situated within a larger oval labeled "Cytoplasm", which is further enclosed by a boundary labeled "Membrane". Outside the cell, the "Environment" is labeled. To the right, a bracket groups "Environment", "Cell", "Membrane", and "Cytoplasm" under the heading "System". A "close" button is located at the bottom right of the window.

Five **Substrates**, four **Enzymes**, and one **Complex** constitute the model. Substrates are interconnected via reaction equations. In this model, there are five Reactors. A Reactor indicates a chemical reaction equation in the Michaelis-Menten scheme. In the case of a cultured cell, these reactions would exist inside a Cell, surrounded by an Environment. A Membrane compartmentalizes the external world and the cell interior, which is filled with a Cytoplasm.

close

Done

図33. Model Informationの表示 (Toy Model)

また、コントロールパネルの[Help]メニューから、[Model Information]を選択すると、ブラウザ上にモデルの説明が表示されます。

この節の最後に、シミュレーションの状態の保存方法を説明します。シミュレーションの途中経過を保存するには、コントロールパネル上で“Stop”ボタンを押し、[File]メニューから[Save Cell State]を選択します。

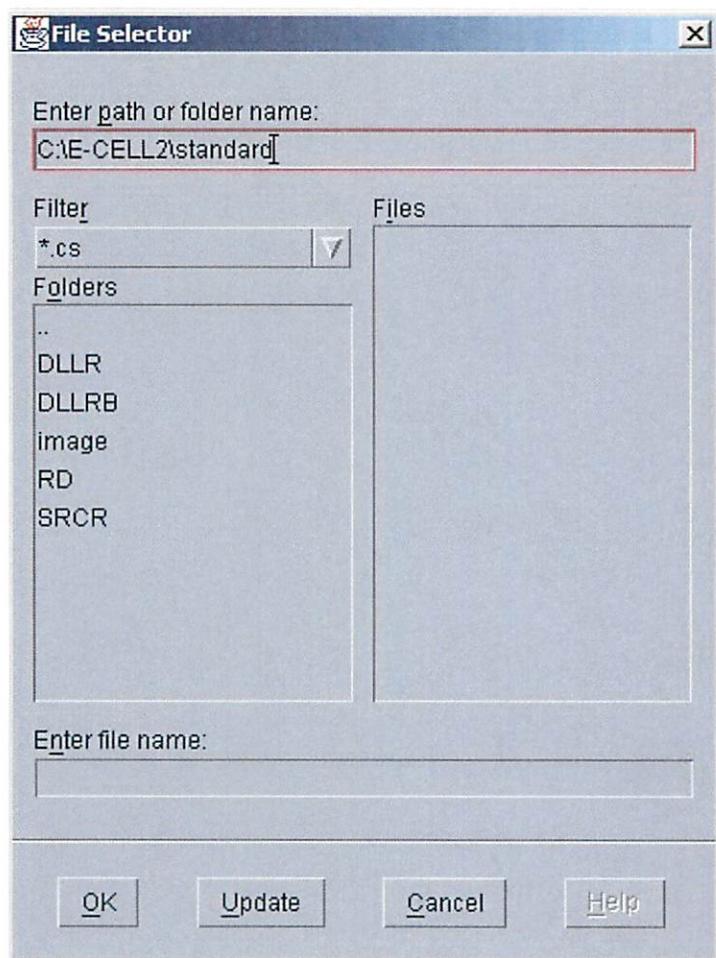


図34. Cell State Fileの保存

細胞モデル全体の状態を記述したファイルは、拡張子”.cs”が付きます。適当なディレクトリ上に、保存してください。この形式のファイルは、コントロールパネルの[File]メニューから、[Load Cell State]を選択することで読み込むことが可能で、保存された状態からシミュレーションを再開することができます。

## スクリプトファイルによる自動化

大きなシミュレーションになってくると、色々な作業をいちいち手動で行うのが大変です。そこでE-CELLでは、“スクリプト”という機構を用意していて、スクリプトの中に記述された命令をE-CELLに読み込ませることで、シミュレーションを自動化できます。このスクリプトファイルには、通常”.ecs”という拡張子で識別されます。

ここでは、スクリプトの書き方について簡単に説明し、E-CELL1とE-CELL2で、スクリプトファイルの形式が異なる部分についても説明します。

まず、以下のファイルを見てください(行頭の数字は行番号です。)。

```

01 TmpDir /temp
02 LoadRule sample.eri
03
04 NewInterface Tracer A_B_C
05 AddTrace Substance:/CELL/CYTOPLASM:SA

```

```
06 AddTrace Substance:/CELL/CYTOPLASM:SB
07 AddTrace Substance:/CELL/CYTOPLASM:SC
08 SaveAt 500
09
10 UpdateInterval 100
11
12 Run 501
13 SaveCellState sample-after500s.cs
14 Stop
```

1行目：一時ファイルを保存するディレクトリを指定します。

/tempはカレントドライブの直下のtempディレクトリを意味し、インストール時に作成されます。

2行目：読み込みたいルール・インターメディエイト・ファイル

('.eri' ファイル) を指定します。

4-8行目：Tracerに関する表示

4-7行目：AddTraceはTracer内でのみ宣言される。

引数にかかれたSubstance、もしくはReactorをTracerの表示に  
加える。引数は、IDだけでなくそれが存在するSystemとTypeを  
含めて示す必要があります。

8行目：このTracer上に表示させた物質を、E-CELL時間の500秒後に

保存します。Tracer内でのみ宣言されます。

9行目：NewInterfaceの終わりは、空白行が必要になります。

10行目：Tracerに表示させることを100ステップごとに行います。

12行目：501秒間シミュレーションを行います。

13行目：細胞モデルの状態を' sample-after500s.cs' というファイルに  
保存します。

14行目：シミュレーションを停止します。

上記のファイルを、“toy.ecs”のような、適当な名前で保存します(ただし、US-ASCII以外の文字を使用しないで下さい。)。保存するディレクトリは、“sample.eri”がある、“C:\E-CELL2\standard”にしましょう。これ以外のディレクトリにE-CELL2をインストールしている場合は、適当に読み替えてください。

E-CELL2の標準リアクターモデル版を起動しましょう。起動したら、コントロールパネルの[File]メニューから、[Load Script]を選択し、作成した".ecs"ファイルを読み込みましょう。

それから、E-CELL2から実装された"Logger"という機構についても紹介しておきます。Loggerは、細胞の状態をシミュレーション中に随時記録する機構です。細胞の状態をスナップショットで保存するよりも、先進的で、スクリプトファイルに以下のような記述を追加することで利用できます。

```
NewInterface Logger test-log
AddTrace Substance:/CELL/CYTOPLASM:SA
AddTrace Substance:/CELL/CYTOPLASM:SB
AddTrace Substance:/CELL/CYTOPLASM:SC
AddTrace Substance:/CELL/CYTOPLASM:SD
AddTrace Substance:/CELL/CYTOPLASM:E_bc
AddTrace Substance:/CELL/CYTOPLASM:C_Ebc-D
AddTrace Reactor:/CELL/CYTOPLASM:E_ab-0
AddTrace Reactor:/CELL/CYTOPLASM:E_bc-0
AddTrace Reactor:/CELL/CYTOPLASM:E_cd-0
```

最初に作成した、".ecs"ファイルの9行目の後にLoggerに関する記述を追加して、E-CELL2に読み込ませてみましょう。

以下は、Loggerの特徴です。

1. カレントディレクトリーに、"Logger"というディレクトリーができ、そこに、Tracerの"SaveAt"機能で作られるのと同様の書式で、SubstanceまたはReactorごとのファイルが記録されます。
2. 実行中随時ファイルに記録するので、もしE-CELL2がクラッシュしても、記録が残ります。
3. 仮想記憶機能を使わないので、Tracerによる記録より高速です。

シミュレーションが停止したら、Loggerディレクトリ以下の".ecd"ファイルをメモ帳(notepad)などで開いてみましょう。ファイル名が個々のSubstanceやReactorを示していて、"時刻"、"瞬間数"、"平均数"、"最大数"、"最少数"、"瞬間濃度"、"平均濃度"、"最大濃度"、"最小濃度"の情報が記載されています。

ここまでで一通り、シミュレーションをどのような手順で進めていくかを説明しました。個々の項目の詳細な説明については、3章以降を参考にしてください。

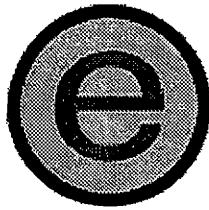
---

E-CELL2 User's Manual

 Last Update \$Date: 2002/11/25 \$.  
Copyright: Keio University and Mitsui Knowledge  
Industry Co., Ltd. 2002

2章 チュートリアル

[1章:3章:4章:5章:6  
章][トップ]



## E-CELL 生物学の講義 第一回

---

1. E-CELL Simulation Environment とは
  2. E-CELL Project とは
  3. E-CELL Project の背景
  4. 細胞シミュレーションの意義
  5. E-CELL 生物学研究プロトコル
- 

### E-CELL Simulation Environment とは

E-CELL Simulation Environment は、分子レベルの全細胞シミュレーションを可能にするためのシミュレーション環境です。E-CELL の名称は、Electronic Cell から来ています。E-CELL の “E” は、e-mail や e-business などの “e” と同じ由来です。つまり、E-CELL とは電子化細胞、コンピュータ上の細胞のことです。E-CELL Simulation Environment を用いることにより、細胞内の化学反応による分子数の変化のシミュレーションを行うことが出来ます。現在、慶應義塾大学生命情報研究室で開発が行われており、version 1.0β が <http://www.e-cell.org> にて公開されています。

分子レベルのシミュレーションとは、反応速度論的な数式を利用することにより時間軸での分子濃度の変化を計算するシミュレーションのことです。E-CELL Simulation Environment では、その仕組みがさらに拡張されており、細胞の構造や、濃度以外の分子の属性等についても表現することができます。

### E-CELL Project とは

E-CELL Project の目的は、全細胞シミュレーションを可能とするモデリングシミュレーション技術を構築することです。全細胞シミュレーションとは、細胞内の全ての反応を網羅したシミュレーションのことです。細胞内の各要素は複雑に相互作用しあうために、部分に注目しただけでは把握できない現象が存在し得ます。そのため、細胞全体を対象としたシミュレーションを行う必要があるのです。

全細胞シミュレーションを可能にするためには、モデリングシミュレーション理論の構築、シミュレータの開発、細胞モデルの構築、モデルやシミュレーション結果の解析手法の構築、データマネジメント手法の開発等を同時にしていく必要があります。E-CELL Project では、これらの研究を同じプロジェクト内で行うことによって各領域間の共同研究を密に行う環境作りを目指しています。

E-CELL Project には、現在教員・学生等、合わせて24人が所属しています。現在行われている研究、または、行う予定の研究には以下のようなものがあります。

- E-CELLバージョン2の開発
- E-CELLバージョン3の開発

- E-CELLバージョン4の開発
- E-CELLバージョン3のGUIの開発
- E-CELL Managerの開発
- 赤血球のモデリングとシミュレーション
- 赤血球の細胞モデルを用いた病態解析
- 遺伝子発現系標準モデルの開発
- 大腸菌lacオペロンのモデリングとシミュレーション
- 全ミトコンドリアモデルのシミュレーション
- 大腸菌化学走性のシグナル伝達経路のモデリングとシミュレーション
- 概日リズムのモデリングと確率論
- 決定論シミュレーション
- 細胞周期のモデリングとシミュレーション
- C3、C4植物の光合成のモデリングとシミュレーション
- 糖尿病のモデリングとシミュレーション
- Substance Reactorモデルの構築
- 拡散系のモデリング手法の構築
- パラメータ推定手法の構築
- 反応経路推定手法の構築
- 定常状態の程度の数値化手法、計算手法の構築
- 時系列解析手法の構築

E-CELL Project の成果は *Bioinformatics* 誌に掲載されました。また、サイエンス誌、ネイチャー誌、日本経済新聞などで紹介されています。

## E-CELL Projectの背景

### 歴史的背景

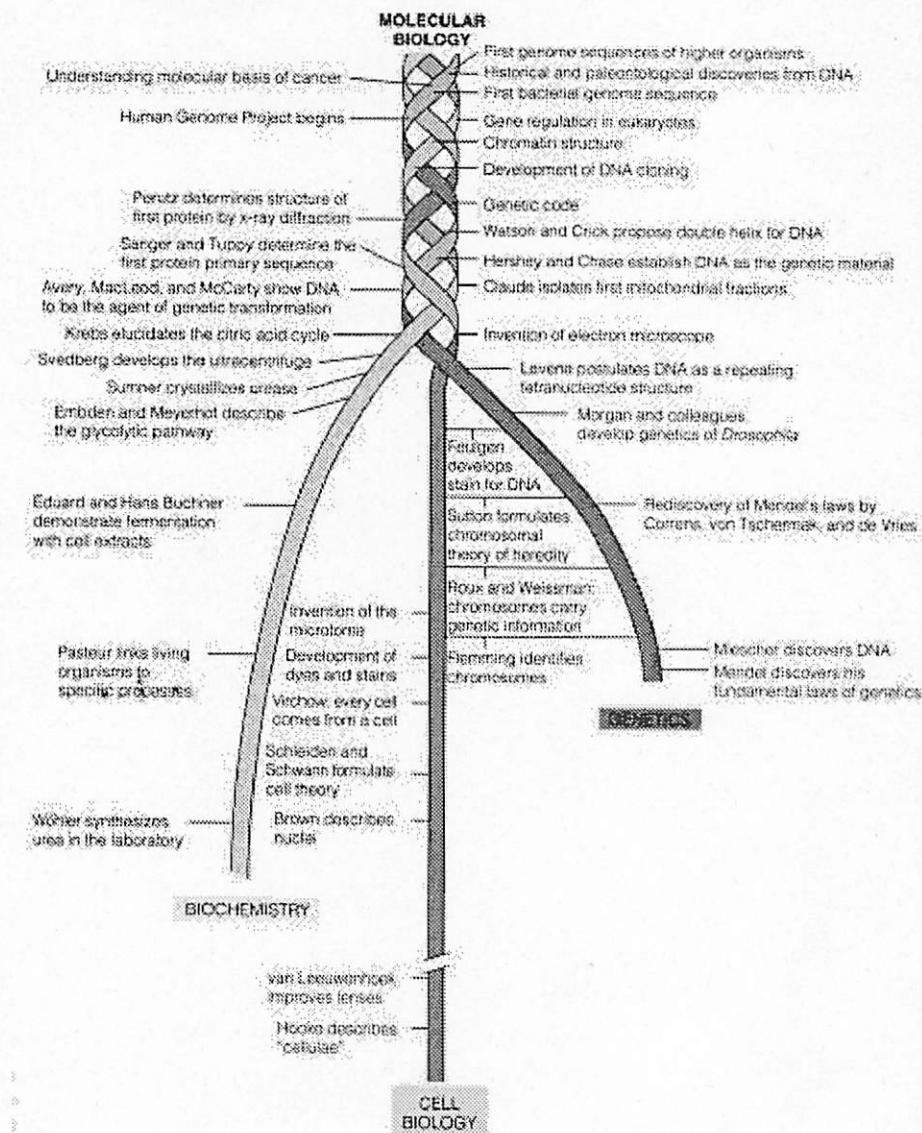
### 分子生物学

E-CELL project が前提としている主要な知識体系の一つは、「分子生物学」という学問分野に属するものです。分子生物学とは、生命現象を分子のふるまいとして記述する学問です。生物学の最先端として、まあ、なんていうか、かなりホットな領域なんです。日々新しい発見があり、将来的にもたくさんの発見が生まれると期待されています。

E-CELL のモデルは その知識によってたっており、かつ、その分野における貢献を期待されています。そこでまず、分子生物学に至るまでの生命探究の歴史をお話します。

分子生物学は一般に、細胞生物学と生化学と遺伝学の融合といわれています。生命現象を対象にしてさまざまな学問分野の人々が研究してきたことが、分子生物学というひとつの枠組にまとまりつつあるということです。

初期の生物学は、あらゆる生物種を観察、記録して ひたすら分類する学問でした。それが、いくつかのブレークスルーを経て現在の分子生物学という体系をもつ科学になるまでの流れは、以下のようにになっています。



生物を対象にした学問の流れ(Mathews "Biochemistry" より引用。後でかえます)  
それらの分野について簡単に説明します。まだ途中です。ごめん。

### • 細胞生物学

生物の構造や構成物質とそれらの相互作用の理解は、顕微鏡の技術発展とともに進みました。「全ての生物に共通する」といえる構造が発見されます。それが細胞です。これまで、「全ての生物には～がある」という言い方でしか語れなかつたものが、「全ての生物は細胞からできている」と言えるようになったのです。いまでは、細胞の内部まで細かく調べられ、遺伝子の本体が DNA という物質であること、その構造までもつきとめられています。

### • 生化学,生物物理学

生物と無生物の違いはどこにあるかという問い合わせに対し、生物は物理法則とはちがう性質の力に支配されていると考える人が多くいました。それは、生体と切りはなして行なった実験により、段階を追って否定されてゆきます。生化学の始まりです。

生物の構成物質(有機物)は、無生物界の物質(無機物)とは根本的に違うもので、物理法則とも関係ない、何か別のものである、という考えがありました。それは、無機物である ammonium cyanate から、腎臓で作られる尿素が実験室で合成されたことで否定されました。

それでも、生体内でおこっている反応は、物理や化学の法則の外にある何か特別な力によっておこっているのではないかという反論がありました。しかしそれも、破壊されたイーストの細胞からの抽出物が、発酵というプロセスを生体内ではなく試験管の中でも行なえることが示されました。

生命の実体は ただの物質で構成されていて、生命現象は既存の物理法則に従っているのだ、というコンセンサスができてからは、生命活動を対象に、物理学や化学の理論を適用してその性質を探る手法が発達していきました。

- 遺伝学

メンデルにより遺伝を司る因子の存在が示唆され、のちに証明された後は、その因子の本体が探されました。

- 分子生物学

遺伝子の本体は DNA であるとつきとめられ、その構造が二重らせんであることまで解明されました。その後、この三つ分野は互いに密接にかかわりながら研究されています。

## 計算機科学

生命科学が物理科学と異なる点に、現象の記述の蓄積が特に重視されることがあります。これは、生物と生命現象の多様性と個別性による所が大きいと思われます。今まででは、現象があまりにも多様で知識が膨大なので、知識の蓄積から意味のある情報をみつけることや原理を導くことは他の学問にくらべて困難で、時間のいるものでした。

しかし、近年のコンピュータの計算能力の向上につれて、膨大な情報の整理が容易になりました。「バイオインフォマティクス」はそうしたデータ処理技術としてはじまり、現在はそれにとどまらない知識発見の手段に発展しつつあります。

生命という複雑なシステムの本質の理解へむけて、計算機という道具を得てより多様なアプローチが試みられるようになってきました。その具体的なアプローチのひとつが、シミュレーションを軸にした研究手法、つまりあとで説明する E-CELL 生物学なのです。

## ゲノムプロジェクトとの関係

E-CELL Project の背景をさらに、もう一つ違った角度から眺めてみようかと思います。皆さんご存知のとおり、今、世界中でゲノムプロジェクトというものが進行しており、まさに一段階目の終わりを迎えるようとしています。ゲノムプロジェクトとは、一つの生物のゲノム配列を決定するものです。生命体の構造、機能を司るのは遺伝子です。全ゲノム配列の決定には、その生物の持つ全ての遺伝子セットを解明すると言う意味があります。つまり、その生物の設計図の全セットを手に入れるということです。数ヶ月前に、アメリカで、人のゲノム配列はほぼ読み終わったとの発表もありました。また、現在30種類近い微生物、動物についての全ゲノム配列がすでに決定しています。今後は、そのゲノム配列の情報を元にどのようにそれを役立てて行くかということが、研究の中心となって行く時代に入ったわけです。

ゲノムプロジェクト後の研究として現在行われているものにはいくつかの流れがあります。それらの主な特徴は、あるレベルでのその生物の機能を網羅的に解明しようとするものです。

その一つはトランскルiptーム解析と呼ばれるもので、一つの生物の遺伝子の発現状態を網羅的に調べようとするものです。さらに、遺伝子の発現状態の時間変化の情報を元に、遺伝子同士の相互作用を調べようとする研究もあります。

また、プロテオーム解析と呼ばれるものもあります。これは、ゲノムの情報をもとに一つの生物のタンパク質の構造や機能を調べようとするものです。さらには、タンパク質間の結合可能性を調べることにより、タンパク質同士の相互作用を網羅的に調べようとする研究もあります。

トランスクリプトーム解析は遺伝子発現について、プロテオーム解析は主にシグナル伝達経路についての事実が多く解明されるのに対し、細胞内の反応と反応同士の相互作用を網羅的に調べることにより、細胞内代謝についての解析を行うのが、メタボローム解析です。

これらの網羅的な研究結果により、生命体を全体として解析するための準備が整いつつあると言えるでしょう。要素の詳細な解析、及び、要素同士の個別の相互作用が調べられたら、次は、その相互作用の作るネットワークについての研究を行う段階となります。その最も有用だとされているのが、構成的手法です。これは、判明している要素を組み合わせ、必要ならばそこに仮説を加え、実際の現象と比較することにより、未知の事実を探ろうとするものです。コンピュータによるモデリングとシミュレーションはこの構成的手法の中心となるものです。

## 細胞シミュレーションの意義



細胞シミュレーションの意義は3つに分けて考えられます。一つ目は、*in silico* 実験を可能とすること、二つ目は、構成的手法であるということ、三つ目は情報科学、統計力学の手法が使えるということです。

### *in silico* 実験

現在、生命科学の分野では、様々な先進的な研究が行われています。この分野の成長により、学術的、医学的、産業技術的な様々な飛躍が期待されています。学術的には、生命体の統合的理 解、医学的には疾患の新しい診断、治療、予防法の開発、新しい創薬の試み、さらにはオーダーメード医療等、産業技術としてはバイオテクノロジーによる工業、農業、畜産の新技术の開発、食料・環境問題の解決など、様々な領域での発展が期待されています。

ところで、生命科学による生命現象の理解のためには、実験による証明が不可欠です。実験には、*in vivo*、*in vitro* の二通りのものがあります。前者は細胞内、後者は試験管内の意味です。*in vivo* の実験はより正確ですが、より困難なものとなり、*in vitro* の実験はより容易に行うことが出来ますが、細胞内の場合と挙動が違う可能性もあります。しかし、いずれにしても、実験は時に非常に効率が悪く、手間、時間、金銭がかかります。そこで、コンピュータシミュレーションによりより効率よく実験の予測を行おう、というアイデアがあります。これを、*in silico* 実験と言います。「*in silico*」は、「計算機内の」という意味です。*in silico* 実験では、より効率よく実験を行うことができるというだけではなく、計算機技術により、より有用な実験の条件を自動的に探索させたり、全通りの条件の実験を同時に実行等のことも可能となります。

このため、今後、細胞シミュレーションは、生命科学の発展を支えて行く重要な技術となることが予測できます。

### 構成的手法による生命の普遍的原理の探求

生物学の基本的な命題の一つに、「生命とは何か」というものがあります。物質と生命との違いとは何か、生命の本質である普遍的な原理とはなにかということです。この問い合わせるべくこれまで様々な研究が行われてきましたが、この答えは未だ得られていません。歴史的に、生命を形作る物質の中にその本質があるのではないかとされ研究されてきましたが、現在のところ、生命体は物質の集まりで出来ているという以上のことは解明されていません。

そんな中で、生命原理とは、ゲノムの情報を基に物質が集まって出来た高度に秩序化され、階層化された相互作用のネットワークにこそ、潜んでいるのではないかと考えられるようになってきています。この視点の下に生命を理解するためには、これまで続けられてきた要素還元的な手法だけでは不足です。そして、現在、構成的手法が、生命を理解するための鍵になると考えられています。つまり、分子間の反応にまで還元した要素と相互作用を、再び統合し再構築し、実際の生命体との比較を行う手法、つまり、コンピュータシミュレーションこそが生命原理の解明の可能性を強く秘めていると考えられます。

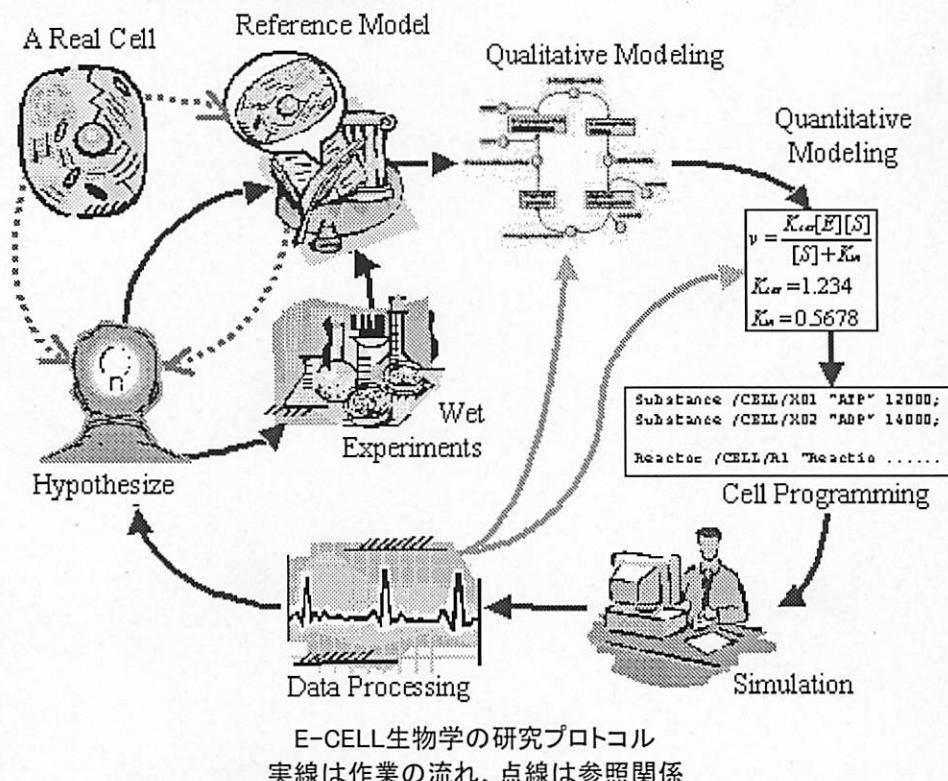
## 情報科学、統計力学の手法

細胞内の現象を解明するためには、仮説を提唱し、実験によって証明するという流れを取ります。仮説の構築は、研究者の思いつきに拘っているところが大きいです。しかし、細胞シミュレーションの手法を使うと、これに加えて情報科学や統計力学の手法が使用できるようになります。つまり、仮説を構築するための方法論と、実行のための仕組みを設計することにより、複雑な現象をうまく説明する仮説を計算により求めることができます。具体的には、代謝経路の中の未知経路の推定や、パラメータの推定などがこれにあたります。

## E-CELL生物学研究プロトコル

全細胞シミュレーションは、それのみでは有用であるとはいえない。既存の生物学と協調して研究を行うことにより、始めて重要な成果が期待できるものとなります。細胞シミュレーションと既存の生物学を統合した手法により新たな現象の解明を目指す学問を「E-CELL 生物学」と呼ぶこととします。

E-CELL生物学における研究プロトコルは、下図のようになります。



まず、左上の“A Real Cell”的図から話を始めます。これは、実際の細胞を表しています。これの持つ特徴、原理を解明するのが目的です。

“A Real Cell”に関して、人間が把握している像を“Reference Model”と呼びます。これは仮説や事実といった知識の集積で、教科書や論文などから探すことになります。“Reference Model”か

ら、シミュレーションの対象となる分子間の相互作用を抜き出して統合することにより、“Qualitative Modeling”が行われます。Qualitativeとは、質的なということです。実験により判明している現象をもとにモデリングを行いますが、時には仮説を立ててモデルに組み込むこともあります。実験結果は、すでにその実験が行われている場合はその内容が記載されている文献を入手するか、データベースからの検索を行います。また、運がよければ、すでにモデリングが行われている場合もあり、統合されたデータを入手することが出来るかもしれません。必要な実験を行う枠組みが用意されていれば、実験を行ってデータを入手することも出来ます。

さらに右に進んで、“Quantitative Modeling”。これは、量的なモデリングのことです。分子間の相互作用を反応速度論的な数式で表現し、速度定数や各分子の濃度の初期値などの数値を決定します。“Qualitative Modeling”と同様、文献などから、すでに解明されている数式や数値を入手します。しかし、数式が入手できなかった場合、Qualitativeな情報を基に、自分で数式を立てることもあります。また、数式で表せない相互作用であった場合は、その詳細な仕組みをデザインしなければなりません。また、数値についてですが、生物学の情報の中で、この種の相互作用に関する数値は最も手に入りにくいものの一つです。数値が手に入らなかった場合は、推測値を指定した後、なんらかの方法により、その推測値を最適化しなければなりません。

次の段階は“Cell Programing”です。ここで、組み上げたモデルをコンピュータが理解できる言語で記述しなおします。通常、人間が記入しやすい形で記入した後に、コンピュータが理解しやすい言語に翻訳されます。

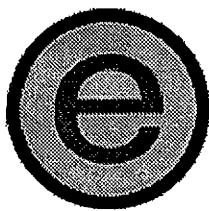
ここまで終了すると、やっとシミュレーションを行うこと(“Run”)が出来ます。しかし、たいていの人工言語によるプログラミングと同様、様々な原因によるバグのせいでシミュレーションが上手く動かず、デバッグという段階を踏む必要があることが多いです。

シミュレーションの結果は時間軸に沿った各数値の変化という形で得ることが出来ます。今度は、この結果を解析(“Data Processing”)します。この解析により得られた解析結果を基に、四通りの流れが考えられます。最初の二つは、解析結果を基にモデルの質的、もしくは量的な性質の変更を行うというものです。この過程、特に量的な性質の変更を自動化し、繰り返すことにより、未知パラメータの最適化等による仮説の構築が行えるようになります。シミュレーション結果の解析後の三つ目の行動は、新しい現象の予測、仮説の提唱です。仮説は、“Reference Model”に組み込まれることとなります。この過程には大抵、*in vivo*, *in vitro* 実験 (“Wet Experiments”) による検証が含まれます。

E-CELL生物学の対象はこの図で示した全ての実線部分であり、その目的はそれについての法論を構築することです。

---

Last modified: Tue Oct 10 14:42:18 GMT 2000



## E-CELL 生物学の講義 第二回

---

1. E-CELL Web Page の紹介
  2. バイオシミュレーション概説 シミュレーションには、様々なモデリング法があります。どのモデルに基づいてシミュレーションを行なうかは、対象の性質やシミュレーションの目的により選択します。今回の講義では細胞のモデリングに関するモデリング法を紹介し、それぞれの手法の視点から E-CELL のモデリングについても解説します。
    - 空間: 均一系と非均一系
    - 反応モデル: 決定論モデルと確率論モデル
    - 常微分方程式と偏微分方程式
    - 代数式と微分方程式を用いたモデリング・シミュレーション
    - 一細胞モデルと多細胞モデル
  3. E-CELL 以外のシミュレーションモデル
- 

### E-CELL Web Page の紹介

1. <http://www.e-cell.org>

E-CELLプロジェクトのトップページ。

- What's E-CELL Project?  
E-CELL Projectとは。
- Articles on E-CELL in Science and Nature.  
Science, Nature誌に掲載されたE-CELLの紹介。
- Posters and Papers  
過去に発表されたポスター、論文。
- Models  
構築されたモデルの公開。現在、自活細胞モデルのみ。
- Software  
E-CELL Simulation Environmentの開発に関するページ。ソフトウェアのダウンロードや、メーリングリスト、ユーザマニュアル等も。
- Members  
メンバー紹介。

2. <http://wg.e-cell.org> , <http://tf.e-cell.org> , <http://eg.e-cell.org>

E-CELLプロジェクト内のメーリングリストのページ。アクセスにはユーザ名、パスワードの入力が必要。グループへの参加、脱退、過去ログの参照などができる。

内容によって3種類に分かれる。

- WG (WorkGroup) : E-CELL Projectの活動の一環として具体的な成果目標を持つ、論

- 文発表(タームペーパー含む)を目指す場合、WGを立ち上げなければならない。
- TF (TaskForce) : E-CELL Projectの活動の一環として論文以外の具体的な成果目標を目指す場合、TFを立ち上げなければならない。
  - EG (E-CELL Groups): E-CELL Projectのその他の活動で、メーリングリストを用いたほうがよいもの。(勉強会、研究予備調査、戦略会議、ネットワーク管理、さぼり、食生活改善プロジェクトなどなんでも)

## バイオシミュレーション概説

空間: 均一系と非均一系

### 1. 定義

均一系とは、物質が空間内に均一な濃度で存在している系のことです。ある物質の濃度に注目して言えば、その物質の濃度は、どの空間をとっても同じになっています。

非均一系は、それとは違い、ある物質が空間内に局在している系のことです。分子の大きさ、反応の早さ、反応の局在等が、非均一系となる原因となります。

### 2. 必要性

均一系を仮定すると、より少ない計算量で現実に近い細胞モデルをつくることが可能です。細胞内では様々な分子が動きまわっていて、分子どうしが衝突することによって反応が起こります。しかし、分子が小さくて、移動の速度が速い場合、その物質は一瞬にして広がるため濃度は均一になると仮定することができます。濃度が均一なので分子一つ一つの動きを計算する必要がありません。また、空間での濃度変化を計算する必要もないで、計算量を大幅に少なくすることができます。細胞内の現象の多くは均一系と仮定することができます。現在のE-CELLでは基本的に均一系としてモデリングを行なっています。

それに対して、均一系では表現できない場合もあります。たとえば、分子が大きくてゆっくり移動する場合や、ある局在している反応の速度が拡散の速度に比べて十分に速い場合は、場所による分子の濃度勾配ができてしまいます。また、分子数が少ない場合、すべての場所に物質が広がることはできません。実際に、発生時の細胞分化等、細胞内の分子が局在していることで働いている仕組みもあります。このようなときは、非均一系としてモデリングしなければなりません。

### 3. 方法

非均一系のモデリングにはいくつかの方法があります。主な方法として、拡散方程式をつかったり、分子1つ1つの動きを計算する分子動力学をつかったりするのが挙げられます。コンピュータ上で細胞モデルをつくる時に、現実の細胞で起こっているように1つ1つの分子の動きまで考慮すれば、現実に近いモデルをつくることができます。これが、分子動力学を使用する方法です。しかし、分子ひとつひとつの動きをシミュレーションするには膨大な計算量が必要で、現実には大変困難です。それに比べて、分子の濃度の勾配を方程式により導き出して計算しようとするのが、拡散方程式を使用する方法です。

図 均一系と、非均一系の図 図 反応が局在する場合

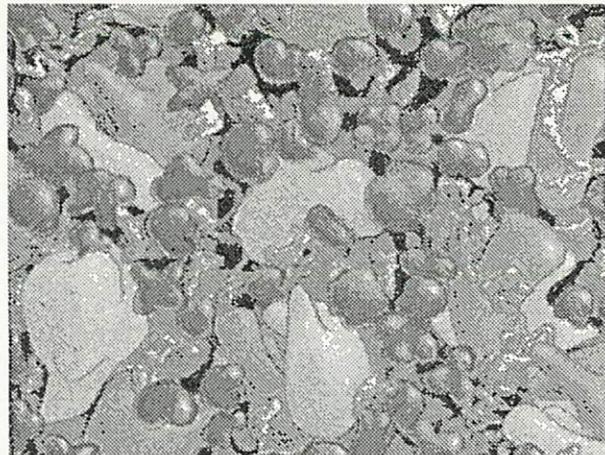


図:macro molecular crowding (細胞質の巨大分子)。「細胞の分子生物学」第3版から引用。巨大分子だけをかいています。青はRNA、緑はリボソーム、赤がタンパク質です。大きい分子は、細胞質を移動中に他の分子と相互作用しやすいといえます。そのため、巨大分子の拡散速度は比較的遅く、小さな分子の拡散速度は水中と同じくらいに速くなっています。

## 反応モデル:決定論的モデルと確率論的モデル

### 1. 定義

決定論的モデルは、ある時点の状態を与えると、その後の全ての時点における系の状態が一つに決まるモデルです。

それに対して確率論的なモデルでは、次の時点で取り得る状態が複数あります。それぞれの状態への移りやすさは確率で表わされます。

### 2. 必要性

反応に関わる分子数の多い場合は均一系を仮定し、決定論的なシミュレーションを行なっても問題は少ないと考えられています。決定論的なシミュレーションは再現性があるため、系の動態の分析や予測が簡単にできます。代謝系と呼ばれる反応の集まりではそれぞれの種類の分子が数多く存在している場合が多いので、決定論的なシミュレーションが適しています。

分子の数が少ないとときは、一個一個の分子の動きが系の振舞いに大きな影響を与えることがあります。これは、サイズ効果と呼ばれます。こういった場合は分子の運動を表現するために確率論的なモデリングが必要となります。遺伝子の発現調節や、シグナル伝達といった制御機構のほとんどは、少数の分子どうしの相互作用によって成立しているため、確率論的なモデリングが有効になります。特に、遺伝子の発現調節では、遺伝子発現の開始をON, OFFで表すことが重要となり、その違いがその後の細胞の挙動に大きな影響を与えます。

確率論的シミュレーションは結果がひとつに決まらないので、再現性がないという点に注意しなければいけません。

### 3. 方法

現在の E-CELL project では、反応の設計方法を変えることにより、ユーザが任意に双方のモデルを選択できます。標準的なものとして用意されている反応では、基本的に決定論的なモデリングを採用しています。通常、反応動力学的な式を用いると、決定論的な結果が得られます。

確率論的なモデリングは、色々な方法があるのですが、基本的に乱数を発生させて計算します。確率論と決定論を対象によって使い分けることが必要となります。

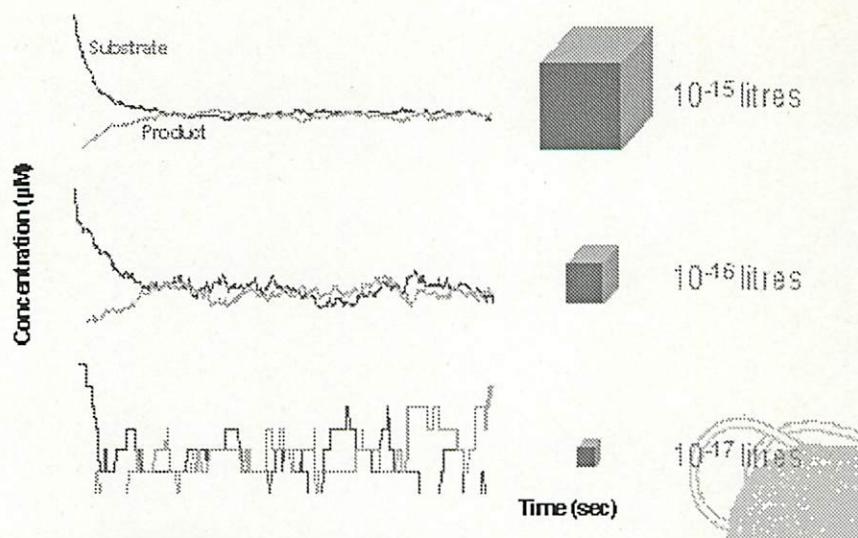


図:分子数の多い場合と少ない場合のグラフ(StochSim のドキュメントより引用)  
それぞれの体積における物質の濃度変化のグラフです。体積が少なくなるほど、グラフが滑らかではなくなります。少ない体積下で分子の数が少ない場合、平均化した濃度変化と実際のものとのへだたりが大きくなっていることが読みとれます。

## 常微分方程式と偏微分方程式

### 1. 定義

常微分方程式は、1つの独立変数に対する変化量を表わす式です。

偏微分方程式は、2つ以上の独立変数に対する変化量を表わす式です。

### 2. 必要性

均一系の場合、時間に応じた各物質の変化量を定義すればいいので、系の状態を常微分方程式だけで表現することができます。一方、非均一系として拡散を取り入れる場合は、偏微分方程式も必要となります。時間に対する物質の変化量だけでなく、場所による変化量も定義しなければなりません。

### 3. 方法

常微分は、ルンゲクッタ法、オイラー法などを使って数値的にタイムステップ毎に計算します。

偏微分は、陽解法と陰解法があって、E-CELLでは陽解法を使っています。

## 代数式と微分方程式を用いたモデリング・シミュレーション

### 1. 定義

微分方程式で表される細胞内の現象は、微小時間当たりの分子数の変量を計算する場合です。

それに対して、代数式で表される細胞内の現象は、ある時間にある代数式が成り立ってい

ることを前提とする場合です。

## 2. 必要性

分子数の変化量を時間の関数として表現できる場合には微分方程式を使用するのが適当です。

時間の関数として表現できない場合には、代数式として表現されます。例えば、迅速平衡の場合などがそれで、この場合、微分方程式の計算に用いるタイムステップよりも短い時間で、反応が平衡状態に達することを前提としています。そのため、全ての時間において、平衡式が成り立つなければなりません。

## 3. 方法

微分方程式により表現される場合は、その方程式を基に微小時間での各分子数(濃度)の変量を求めます。

代数式により表現される場合は、その代数式が成り立つための条件を満たす解を求めることがあります。数式的には、連立方程式を立て、それを解けば良いのですが、非線形連立方程式の場合、解を陽的に求める一般的な手法はありません。そこで、コンピュータプログラミングによる数値計算法を用いることが必要となります。

また、代数式と微分方程式を両方同時に使用してシミュレーションを行うことは非常に難解な問題となります。このような系はDAシステムと呼ばれ、一般的な問題となっています。

E-CELLでは、数値積分を行った各タイムステップの後に、代数式による計算を行うことによってこの問題を回避しています。代数式の解法についてはユーザに任せていますが、効率は悪いのですが多くの場合に対処できる解法が用意されています。

## 4. 例

たとえば、 $A + B \rightleftharpoons C \rightleftharpoons D$  という反応を例にあげると、この反応を微分方程式で表した場合、以下のようになります。

$$\begin{aligned} d[A]/dt &= -k_1 [A] [B] + k_2 [C] \\ d[B]/dt &= -k_1 [A] [B] + k_2 [C] \\ d[C]/dt &= k_1 [A] [B] - k_2 [C] - k_3 [C] + k_4 [D] \\ d[D]/dt &= k_3 [C] - k_4 [D] \end{aligned}$$

この式を基に、微小時間での各分子数の変量を求めれば良いわけです。同じ反応を代数式で表した場合、以下のようになります。

$$\begin{aligned} Kd1 &= [A] [B] / [C] \\ Kd2 &= [C] / [D] \end{aligned}$$

質量作用の法則？

? (この式が成り立たない値を与えられた場合は、この式を成り立たせるような変量を求めなければなりません。それには、以下のような連立方程式を解くことになります。

$$\begin{aligned} [A] &= [A]_0 - v_1 \\ [B] &= [B]_0 - v_1 \\ [C] &= [C]_0 + v_1 - v_2 \\ [D] &= [D]_0 + v_2 \\ Kd1 &= [A] [B] / [C] \end{aligned}$$

$[A]_0$



式の状態(定常状態)は  
向かっていきます？

$$Kd2 = [C] / [D]$$

この連立方程式からv1、v2を消して[A]、[B]、[C]、[D]を、[A]0、[B]0、[C]0、[D]0、Kd1、Kd2で表す必要があります。

## 一細胞モデル・細胞群モデル

### 1. 定義

一細胞モデルは、一つの細胞に注目してその挙動を調べようとするものです。

細胞群モデルは、モデルを多くの細胞の平均としてシミュレーションを行うものです。

### 2. 必要性

より現実に即したシミュレーションを目指すためには、一細胞のモデルが適当です。

しかし、実験から得られる情報の多くは細胞群の平均値として得られます。

### 3. 方法

この場合、分子数は整数で表現され、確率論的な計算を取り入れるのが適当です。分子数は整数值で表現されるべきですが、本来連続的に変化するはずの値を離散的に捕らえていいため、

また、複数回のシミュレーションの結果を基に解析を行うことになります。

この場合、分子の量は実数で表されることとなります。想定する細胞数が十分に多いため、全て決定論的な計算でシミュレーションを行うこともできます。

## E-CELL 以外のシミュレーションモデル

StochSim

StochSim web site

確率論モデルをベースにしたシミュレータとしては、大腸菌の化学走性という現象を対象にした StochSim があげられます。このモデルは、種類にかかわらず全ての分子を別々のオブジェクトとして扱います。シミュレーションオブジェクトは、速度定数やその他の物理的要素から算出した 'probability' を参照して相互作用します。この probability というのが、反応をおこす確率です。シミュレーションの計算過程は以下のようになっています。

- ある時間幅においてランダムに2つのオブジェクトを選ぶ
- ランダムナンバーを生成する
- 生成した数字と、選んだオブジェクトの組が対応する 'probability' とを比較して、その反応を起こすかどうかを決定する。
- 反応が起きるとされた場合はその反応を計算する
- システムをアップデートする。

V-CELL

V-CELL Web Site

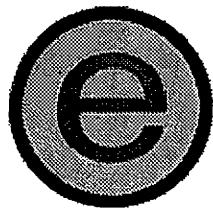
拡散系にも適用できるシミュレータとしては、神経芽細胞のカルシウムウェーブのシミュレーションをはじめとして、汎用シミュレータを目指して現在も開発中の V-CELL があります。

V-CELLは、モデルに物質の位置情報とオルガネラの情報が含まれます。顕微鏡で撮影した映像から空間情報を獲得し、シミュレーションモデルに組みこみます。

拡散する物質には偏微分方程式を、それ以外には常微分を適用して計算します。

---

Last modified: Tue Oct 24 14:53:26 JST 2000



## E-CELL 生物学の講義 第三回

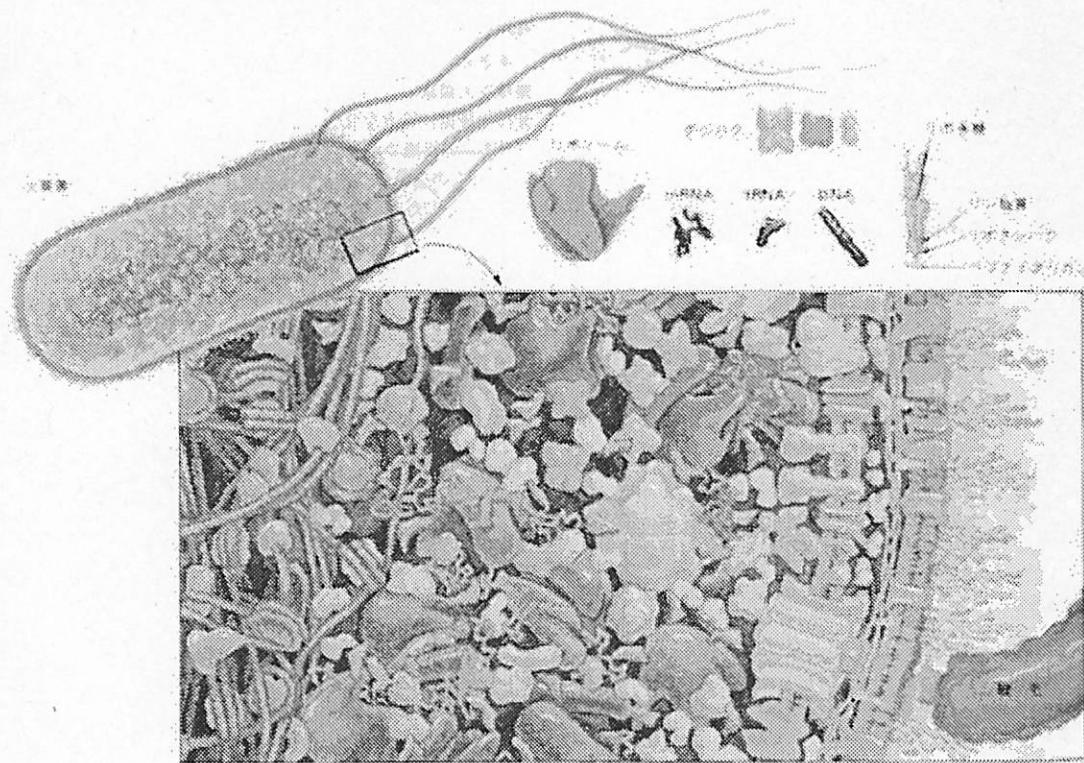
---

1. 細胞ってどんなのよ?
    - 細胞の構造
    - 細胞の機能
  2. 経路図(パスウェイマップ)
  3. 反応速度
    - 反応速度論
    - 反応速度の解析
  4. E-CELL のモデル
    - Substance-Reactor モデル
    - 状態変数について
    - Structured Substance-Reactor モデル
    - Structured Substance-Reactor モデルのオントロジー
- 

### 細胞ってどんなのよ?

細胞とは、すべての生物を構成する単位です。最も単純な生き物は单細胞生物で、一つの細胞で成り立っています。より複雑な生物では、多くの細胞があつまって互いに協調しながら個体を形成しています。

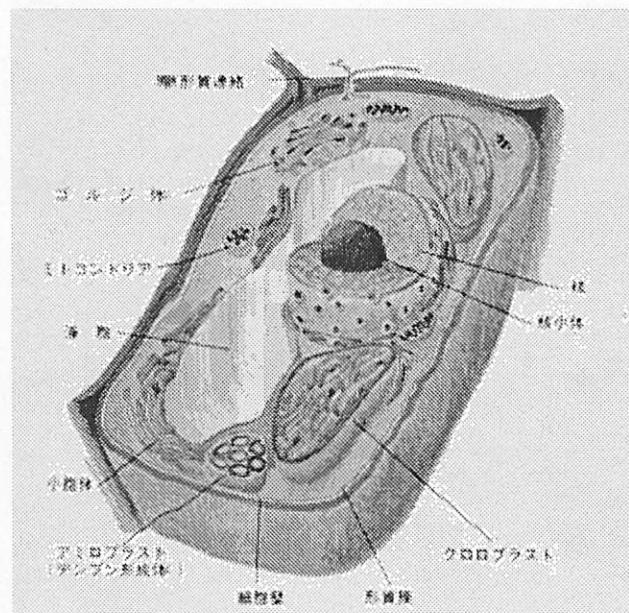
細胞の実体は、無数の物質の濃い水溶液(細胞質)が脂質の膜(細胞膜)によって囲まれたものです。中では様々な分子がひしめきあって相互作用しています。

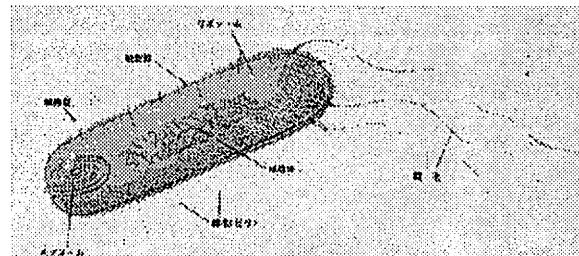


Voet「生化学」より。

## 細胞の構造

下のふたつの図は、真核細胞と原核細胞の構造の模式図です。真核細胞の細胞内には、さらに膜によって囲まれた細胞内小器官があります。特に、染色体の格納されている核があるのが特徴です。原核細胞には細胞内小器官は存在しません。細胞質のなかにいろいろな物質がとけこんでいます。





左:真核細胞の模式図、右:原核細胞の模式図。Voet「生化学」より。

## 細胞の機能

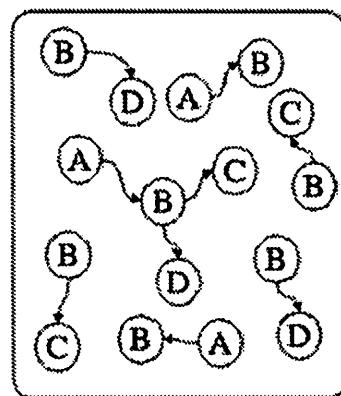
細胞の構成物質は、その相互作用により常に変化しています。物質が、単体あるいは相互に、その原子を組み換えることで異なる物質を生成する現象(化学反応)は、細胞の中で無数に、並行して進行しています。このような、生体内における物質の変化の過程を物質代謝といいます。

これらの化学反応によってたちあらわれる細胞の機能には、以下のようなものがあります。

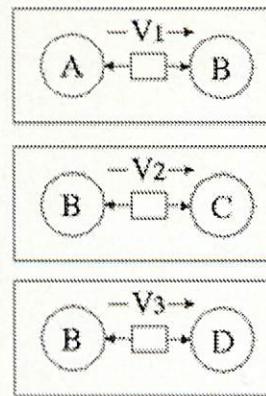
- 特定の生成物に至る一連の酵素反応系(代謝)
- 細胞質中の物質輸送
- 膜を介した物質輸送
- シグナル伝達
- DNA複製
- 遺伝子発現

## 経路図(パスウェイマップ)

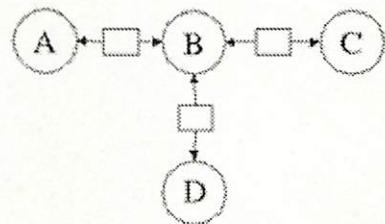
細胞内の反応は下図のように、細胞内のいたるところではばらばらに起こっています。



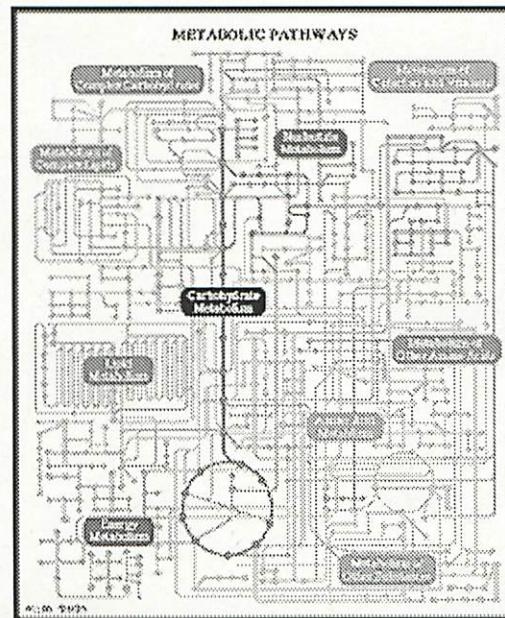
空間(細胞質)の大きさが反応の速さに比べて十分に小さいために均一系を仮定できる場合、上図は、分子種毎にまとめて考えることが可能となります。そして、それぞれの分子種同士の相互作用を抜き出して列挙すると以下のように表せます。



このばらばらの反応を、機能ごとに連鎖反応としてまとめて表したものが経路図です。

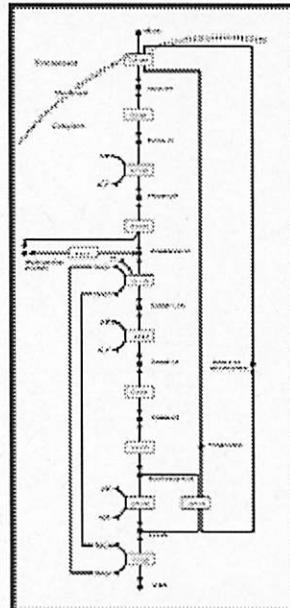


細胞内で進行している化学反応は、いくつかの種類に分類することができます。栄養になる物質の分解や生命活動に必要な物質の合成など、ある特定の生成物に至る一連の酵素反応系を、代謝経路といいます。代謝経路図は、それを物質種どうしの関係の経路図として示したもので、KEGGなどのプロジェクトのページにもまとめられています。KEGGのサイトでは、代謝経路から様々な物質データベースや酵素データベースを参照することができます。細胞内の主な代謝経路は下の図のように膨大なネットワークを成しています。



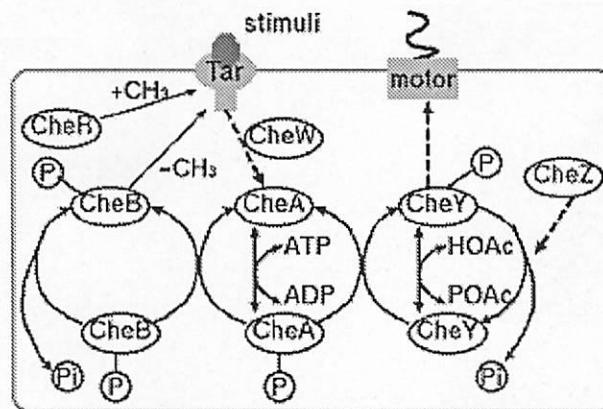
細胞の代表的な代謝経路の概観。“KEGG”のデータベースより  
丸い点が物質、線が物質同士の相互作用(反応)です。

これらの代謝系のひとつを取り出してみるとこのようになっています。

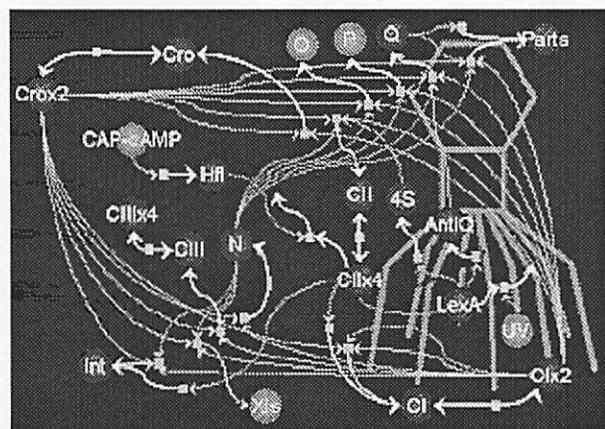


代謝経路の一例(解糖系)  
グルコースを分解してピルビン酸やATP,NADH を生成します。

細胞のその他の機能についても、物質どうしの関係を示した経路図を用いて表現することができます。細胞の様々な機能は物質の相互作用で成り立っているからです。以下にシグナル伝達と遺伝子発現の例を挙げます。



Chemotaxis(シグナル伝達系が含まれる)の反応経路。



遺伝子発現のパスウェイマップの一部。

## E-CELL のモデル

さて、ここまでみてきたように、細胞を、化学系が集合、調和してうごいているシステムであるとみる視点にたつと、ある時点における細胞の状態はその時点における細胞の各成分の量で表すことができます。さらに、細胞というシステムの時間変化は、その系の各反応系の反応速度を求めて次の時点での各成分の量を計算することにより記述できそうです。

しかし、反応の様式には様々な種類があります。もちろん、元をたどれば全て素反応に分解できます。素反応というのは、化学反応において分子、遊離基、原子、イオンに関する結合の変化が微視的な一つの過程で表わされる場合のこと、例えば  $A + B \rightarrow C$  という反応は、 $A + B \rightarrow A$  と  $B$  の複合体  $\rightarrow A$  と  $C$  の複合体  $\rightarrow A + C$  というように分解できます。 $A$  と  $B$  の結合過程、分子内の結合の変更、 $A$  と  $C$  の解離過程という反応過程です。

実際にはそれを全て計算するのは大変だし、反応の途中でできる中間物質の濃度についての実験的データは非常に得にくいなどの要因により、正確なシミュレーションは難しくなります。また、シミュレーションを行う目的により、注目する反応過程の解像度も違い、必ずしも全ての反応素過程の計算を必要としない場合がほとんどです。

### Substance-Reactor モデル

E-CELL では、以下に述べる "Substance-Reactor モデル" という細胞モデリング手法によってシミュレーションを行なっています。

細胞内の反応は、上記したように、均一系を仮定できるならば分子種ごとにまとめて考えることができます。"Substance-Reactor モデル" では、この考え方を基に、細胞を物質(分子種)と反応(相互作用)に分けて表現します。物質により細胞の静態が、反応により細胞の動態が表現されます。

まず、細胞内の分子を、種類ごとに節点(ノード)としてプロットします。質量保存則により、一つの分子種の濃度を減らす現象は必ず別の分子種の濃度を増加させる別の現象と対になります。そこで、化学反応も別の種類の節点としてプロットし、A を減少させて B を増加させることを A からの流束がその減少を表現する節点(化学反応)を経由して B へと流れ込むように書き、係数を経路の重み  $n_{ij}$  として書けば、細胞内の反応系を有向グラフ(向きをもったグラフ)として描くことができます。

(注)

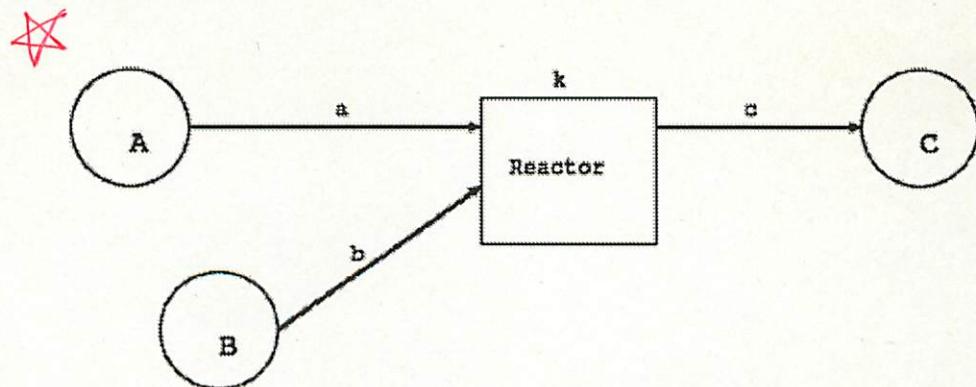
ここででてきた「ノード」「グラフ」という言葉はグラフ理論でよく使われる用語です。グラフ理論は応用数学の一分野で、システムを、点やらと線やらでつないで(つながれたものをグラフと呼ぶ)図示する方法の基礎になっています。

このとき、物質とは別の節点としてプロットした「化学反応」は、E-CELL で「Reactor」として実装しているオブジェクトにあたります。化学反応のように、分子種の濃度を変化させる現象という概念を実体化したのが「Reactor」オブジェクトです。各々の分子種の実体は「Substance(物質)」オブジェクトと定義しています。

(注)

E-CELL はオブジェクト指向にもとづいて構築されています。オブジェクト指向とは、実世界のオブジェクト(もの)になぞらえてソフトウェアを構成する考え方を指します。オブジェクト指向では、実世界のオブジェクトをプログラムのオブジェクトとして表現し、実世界でまとめて考えられるオブジェクトの集合をクラスとしてまとめます。(例: 足、あたま、手(オブジェクト) → 人間(クラス)) 今説明した「Reactor」オブジェクトとは、「Reactor」というクラスのオブジェクトという意味です。Substance-Reactor モデルにおけるこれらの関係は、後述します。

このようにして、「Reactor」に反応の形式を記述して反応速度を計算すると、色々な反応系を扱うことができます。Substance-Reactor モデルは、系の静態を物質の量の集合として表現し、系の動態の様式を、「Substance」と「Reactor」のネットワークのトポロジーとして表現するものです。この動態の表現は、パスウェイマップに似ていますが、化学反応は必ず「Reactor」によって仲介されることに注意してください。例えば、先程反応速度の項で触れた  $aA + bB \rightarrow cC$  という反応は次のように表現されます。



単純な反応の有向グラフによる表記

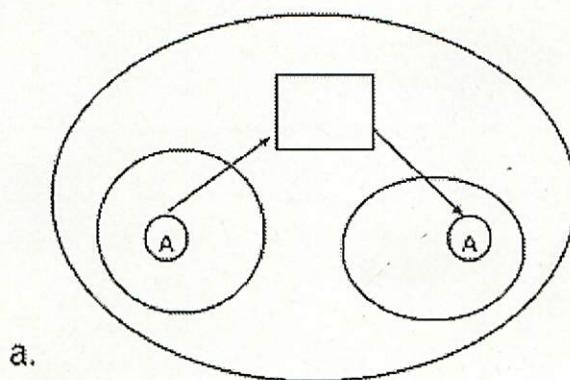
A、B からの流束が「Reactor」を経由して C へ。「Reactor」は、「Substance」オブジェクトである A、B の量をマイナスし、C の量をプラスします。「Substance」オブジェクトを丸で、「Reactor」オブジェクトを四角で表わしています。

### Structured Substance-Reactor モデル

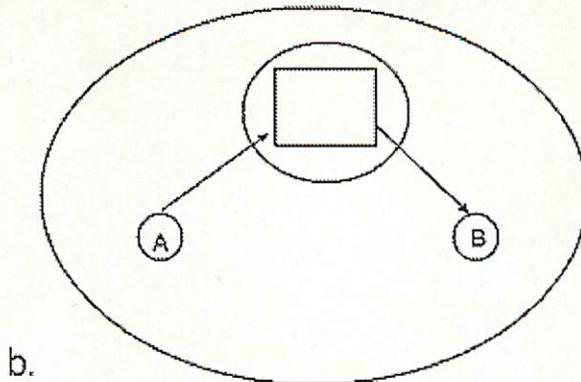
基本的には、Substance-Reactor モデルによってどのような化学反応も表現することができますが、物質の存在する場所を表現する属性間を遷移させるような反応の場合には同じ名前の物質を複数区別する必要があります。この問題を解決するために E-CELLでは「System」クラスを導入することで場所を表現しています。(ここもっとわかりやすく)

Substance-Reactor モデルに「System」を加えた「Substance」「Reactor」「System」の三つのクラスで細胞内の現象を表現するモデルを“Structured Substance-Reactor モデル”と呼びます。「System」クラスのオブジェクトは、三つのクラスのオブジェクトをそれぞれ複数もつことができます。このモデルにより、以下のような現象の表現が可能です。

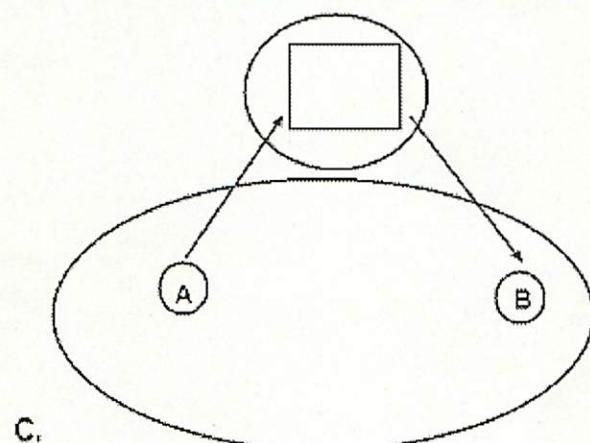
- a. 物質移動。ある「System」から別の「System」への移動を、その両方の「System」を含む「System」(スーパー・システム)にある「Reactor」オブジェクトが仲介。  
(例) なんだろ



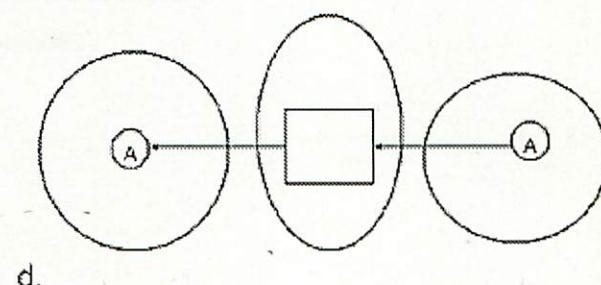
- b. 「System」をまたがった反応。ある反応を表現する「Reactor」がスーパー・システムの物質間の反応を媒介。  
(例) 細胞膜上の酵素に触媒される細胞質の物質どうしの反応



c. 「System」をまたがった反応の別の例。ある「System」上にある  
「Reactor」が直接の上位・下位の関係にはない別の「System」の物質を  
操作。  
(例) 転写、翻訳「System」の「Reactor」が細胞質の物質を操作する場合

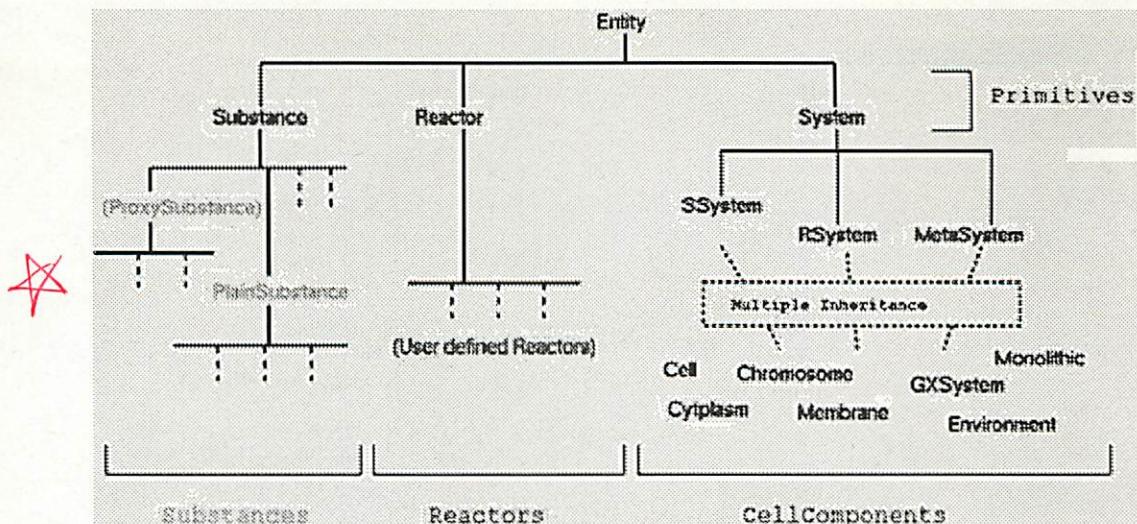


d. ある「System」上の物質が別の「System」上の「Reactor」を介してさらに  
別の「System」へ移動。  
(例) 細胞膜を介した物質輸送



### Structured Substance-Reactor モデルのオントロジー

E-CELLはオブジェクト指向にもとづいて構築されています。オブジェクト指向とは、実世界のオブジェクト(もの)になぞらえてソフトウェアを構成する考え方を指します。オブジェクト指向で作られたソフトウェアには存在論(オントロジー)があります。(ほんと?) E-CELLの、Structured Substance-Reactor モデルによるオントロジーは以下の構造になっています。




---

Last modified: Tue Nov 7 14:27:38 JST 2000



## E-CELL 生物学の講義 第四回

---

1. 導入
  2. 反応速度論
  3. E-CELL内で計算
  4. 定常状態と平衡状態
  5. 酶素反応速度論
    - ミカエリスメンテン式
    - その他の酵素反応
  6. 酶素反応以外の反応
- 

### 導入

前回、代謝経路図(パスウェイマップ)について学習しました。代謝経路は、様々な分子種同士の相互作用の繋がりです。これらの分子種同士の相互作用をSubstance-Reactorモデルに沿って記述することにより、シミュレーションを行うためのルールファイルを作ることができます。

しかし、これだけではシミュレーションを行うことはできません。反応それ自体をどのように計算するかについて考える必要があります。E-CELLでは細胞の活動を各分子種の量の時系列データとして表現しています。そのため、反応として表現されるもの(Reactor)が計算するものは、「分子数の変化する速度」となります。

これを扱うのが、**反応速度論**と呼ばれる分野です。今回は、この反応速度論のうち、E-CELLを使用する上で最低限必要になる基本的な知識について学習します。

E-CELLでは、反応をReactorとしてモデリングします。反応速度式が求まると、その反応についてのReactorを作成することができます。

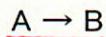
### 反応速度論

前述したように、細胞内では化学反応が平行して進行しています。細胞は、時間と共に変化する化学系の集合であると見ることもできます。化学系とは、一つの化学反応に関連する物質群を外界と区別して呼ぶ呼び方です。

化学系における物質の時間変化を表すのが反応速度です。理化学辞典では、「反応速度」は、「化学反応において反応の原系および生成系に含まれる各成分の時間的変化率」と定義されています。原系とは反応が起きる前の化学系を差し、生成系は反応が起きた後の系を指します。

反応速度に関する問題を扱うのが、反応速度論です。これは物理化学の一部門で、化学系の諸成分の濃度と反応速度との関係や、反応の進行による濃度の時間的変化を調べることから始まりました。測定結果からその反応の速度定数や反応次数などを求めて、反応機構を解明するのが目的です。

ここで、一例として次のような反応を考えてみます。



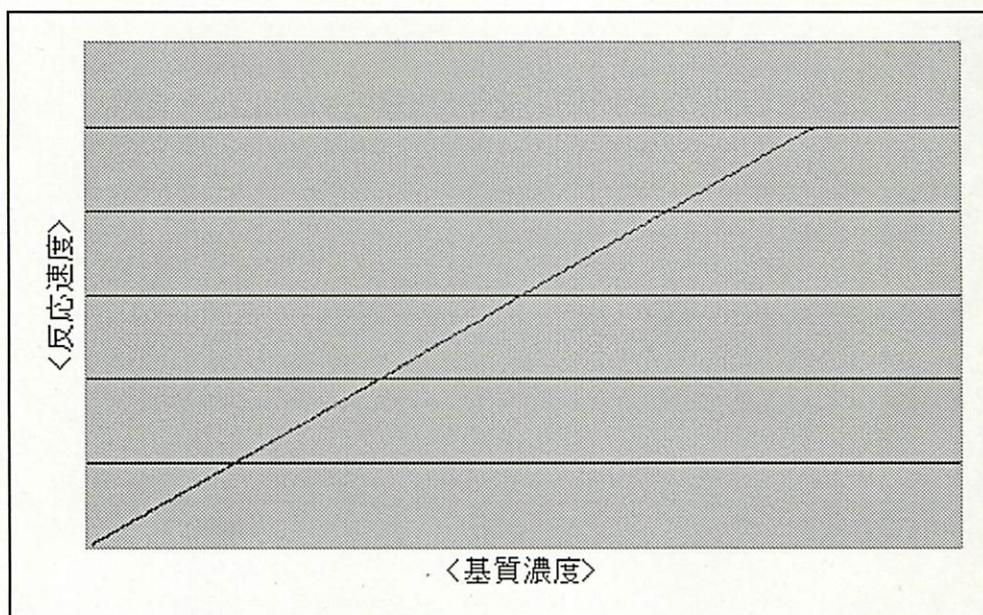
この反応系が閉鎖系で質量保存則に従うとすると、この系の反応速度  $v$  は、以下のように表されます。

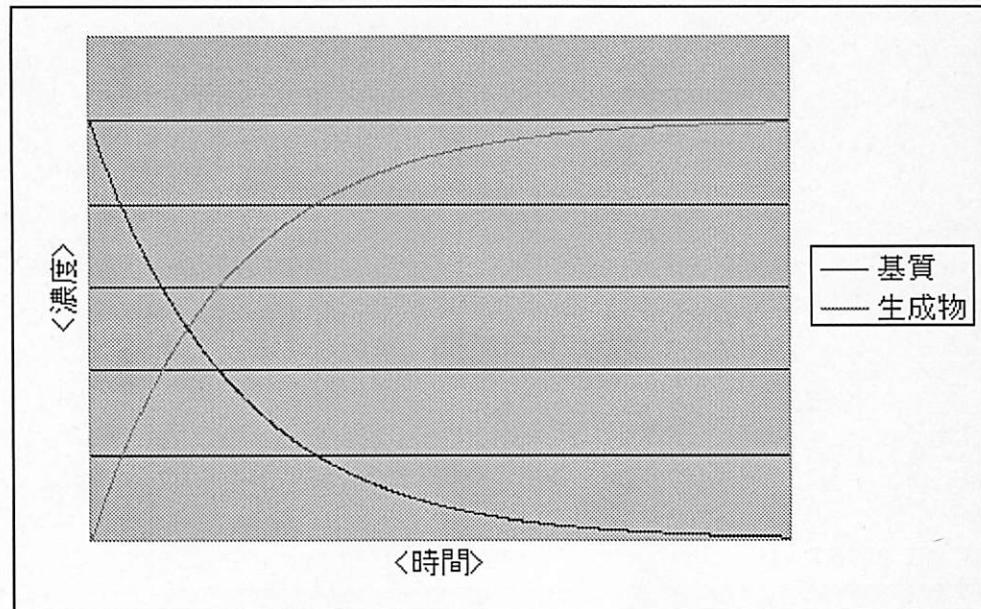
$$v = -d[\text{A}]/dt = d[\text{B}]/dt$$

この反応が触媒の仲介しない一次反応だったとすると、反応速度は基質の濃度(の積)に比例するということが分かっています。基質(substrate)とは、反応前の物質のことです。よって、反応速度は、以下のように表せます。

$$v = k[\text{A}]$$

この反応速度と基質濃度との関係、及びこの反応による基質の時間変化は、グラフにプロットすると以下のようになります。





また、例えば二次反応(二分子反応)だったら、反応速度は以下のようになります。



$$v = -d[A]/dt = -d[B]/dt = d[C]/dt = k[A][B]$$

※これらの、反応速度が基質濃度の積で表される反応のために、E-CELLの標準Reactorとして、MassActionReactorが用意されています。

## E-CELL内の計算

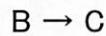
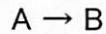
さて、分子種同士の相互作用のルール、反応速度が決定されたら、それをもとにE-CELL内ではどのような計算が行われるのでしょうか。

通常、反応速度は一秒当たりの量で表現されますが、まず、それを積分幅(1タイムステップの時間)当たりの量に変換します。そして、それを各物質の変量として記録します。その各物質毎の変量には、複数の反応の反応速度の値が加算されていきます。そして、全ての反応についての計算が終わったら、最終的な変量を物質の量に反映します。この過程は、数値積分と呼ばれます。

簡単な例をあげましょう。下のような代謝経路図で表されるモデルがあったとします。



これは、二つの反応式に分けることができます。



$$v_1 = \frac{d[A]^2}{dt} = k_1[A]$$

A、B、Cの初期値が1000, 0, 0だったとしましょう。また、一つ目の反応の反応速度定数が $k_1 = 100(1/s)$ 、二つ目の反応の反応速度定数が $k_2 = 50(1/s)$ だったとします。また、一つ目の反応の反応速度を $v_1$ 、二つ目の反応の反応速度を $v_2$ とします。さらに、積分幅を0.001秒とします。

速度は変化する。

すると、E-CELLでの計算は以下のように進んできます。

ステップ	A	B	C	v1	v2
1	1000	0	0	100000	0
2	900	100	0	90000	5000
3	810	185	5	81000	9250
4	729	257	14	72900	12850
...	...	...	...	...	...

といった具合です。これらの数値は、トレーサー等のGUIによって確認することができます。

## 平衡状態と定常状態

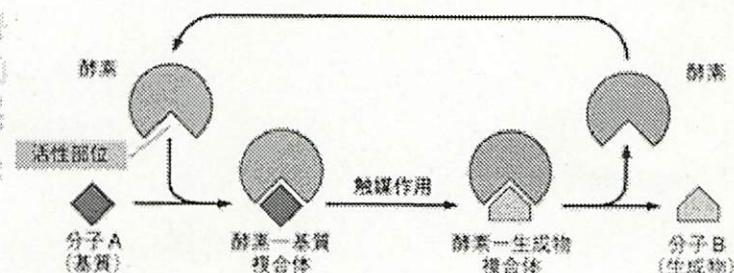
細胞内の、全体、またはある系の状態として平衡状態と定常状態というものがあります。この二つは重要な概念なのですが、混同しがちなので、それについて簡単な説明をします。

平衡状態とは、ある反応が可逆反応だったとして、正方向と逆方向の反応速度が同じである状態ということです。可逆反応とは、正方向、逆方向の両方向の反応が起こりえるということです。平衡状態は、関連する分子の自由エネルギーによって物理的に決定され、基質濃度の積が生成物濃度の積に比例するという関係が成り立ちます。可逆反応は平衡状態に向かって進行します。常に、平衡状態に近い状態を保つような早い可逆反応を迅速平衡反応と呼びます。この場合、E-CELLでは、積分幅よりも短い時間で平衡状態に達する反応として定義します。

定常状態とは、流束(flux)がありながら、物質の量が時間的に不变である状態のことです。流束とは、反応による分子の流れの速度のことです。物質の生成量と消費量が釣り合っている場合にこの状態が成り立ちます。

## 酵素反応速度論

細胞内の代謝経路を構成する反応のほとんどは酵素を触媒として行われます。触媒とは、反応の起こる確率を高め、反応の前と後で量が変化しない物質のことです。ただし、反応の起こる確率を高めるというのは反応の速度を速めるということであり、通常起こらないような反応は、触媒によっても起こることはありません。例えば、通常の反応は平衡状態に向かいいますが、平衡状態から離れるような反応は、酵素が触媒したとしても起こりません。他の自然現象と生命現象との違いを生む大きな要因の一つが、この酵素の存在です。酵素という触媒が高密度で存在することによって、通常起こりにくい反応を選択的に起こすことが出来るのです。



酵素は、基質特異的に働き、加水分解、重合、基転移、酸化還元、脱水、異性化など、様々

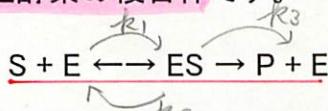
な反応を触媒します。また、ある種の決まった分子により、阻害、活性等の調節を受けます。調節の受け方には様々なパターンがあり、パターンごとに違った挙動を見せます。また、一部の特殊な酵素を除いて、酵素が触媒する反応は、反応速度に上限値を持ち、基質濃度がある程度以上になると、反応速度が一定となるという性質も持っています。

## ミカエリスメンテン式

酵素反応速度論の最も単純なモデルとして、ミカエリスメンテン式により表される反応があります。この反応の反応速度式の求め方を例として説明します。

この反応は、基質、生成物がそれぞれ一種類のみであり、 $V_{max}$ (最大速度)と $K_m$ (ミカエリス定数)の二つの定数により定義されます。

この反応の反応構造は以下のようになります。Sは基質、Eは酵素、Pは生成物、ESは基質と酵素の複合体です。



**使用上の注意** ミカエリスメンテン式は、式を簡略化するためにESについては定常状態が成り立っているという仮定を置いています。また、逆反応を定義しないため、生成物の濃度が非常に高い場合はミカエリスメンテン式により反応速度を求めることはできません。これは、この式が定常状態から離れるような動的なシミュレーションには向かないということを意味します。

ミカエリスメンテン式の反応の素反応ごとに微分方程式を求めるとき、以下のようにになります。

$$\left. \begin{array}{l} d[S]/dt = -k_1[E][S] \\ d[E]/dt = -k_1[E][S] + k_3[ES] \\ d[ES]/dt = k_1[E][S] - k_2[ES] - k_3[ES] \\ d[P]/dt = k_3[ES] \end{array} \right. \begin{array}{l} \cdots (1) \\ \cdots (2) \end{array}$$

求めるべき反応速度( $v$ )は、生成物の生成速度です。これは、(2)式で表されています。しかし、[ES]の量は、反応の中間複合体であり、E-CELLではモデルには組み込まない物質です。また、実験的にも通常求めることができないものです。E-CELLでモデルに組み込み、また、実験で求めやすいのは、全ての酵素の量の合計です。そのため、[ES]を、酵素の全体量と基質の量で表現しなければなりません。

酵素の全体量を $[E]_t$ とおきます。すると、以下の式が成り立ちます。

$$[E]_t = [E] + [ES] \quad \dots (3)$$

また、上記の仮定(ESについては定常状態が成り立っている)により、以下の式が成り立ちます。

$$d[ES]/dt = 0 \quad \dots (4)$$

(1)式に(3)式、(4)式を代入すると、以下のようになります。

$$k_1([E]t - [ES])S = (k_2 + k_3)[ES]$$

「ES」について解くと、

$$[ES] = [E]t[S]/(K_m + [S])$$

ただし、 $K_m = (k_2 + k_3) / k_1 \dots (5)$

となり、[ES]を、酵素の全体量と基質の量によって表現することができました。この式から、反応速度は、

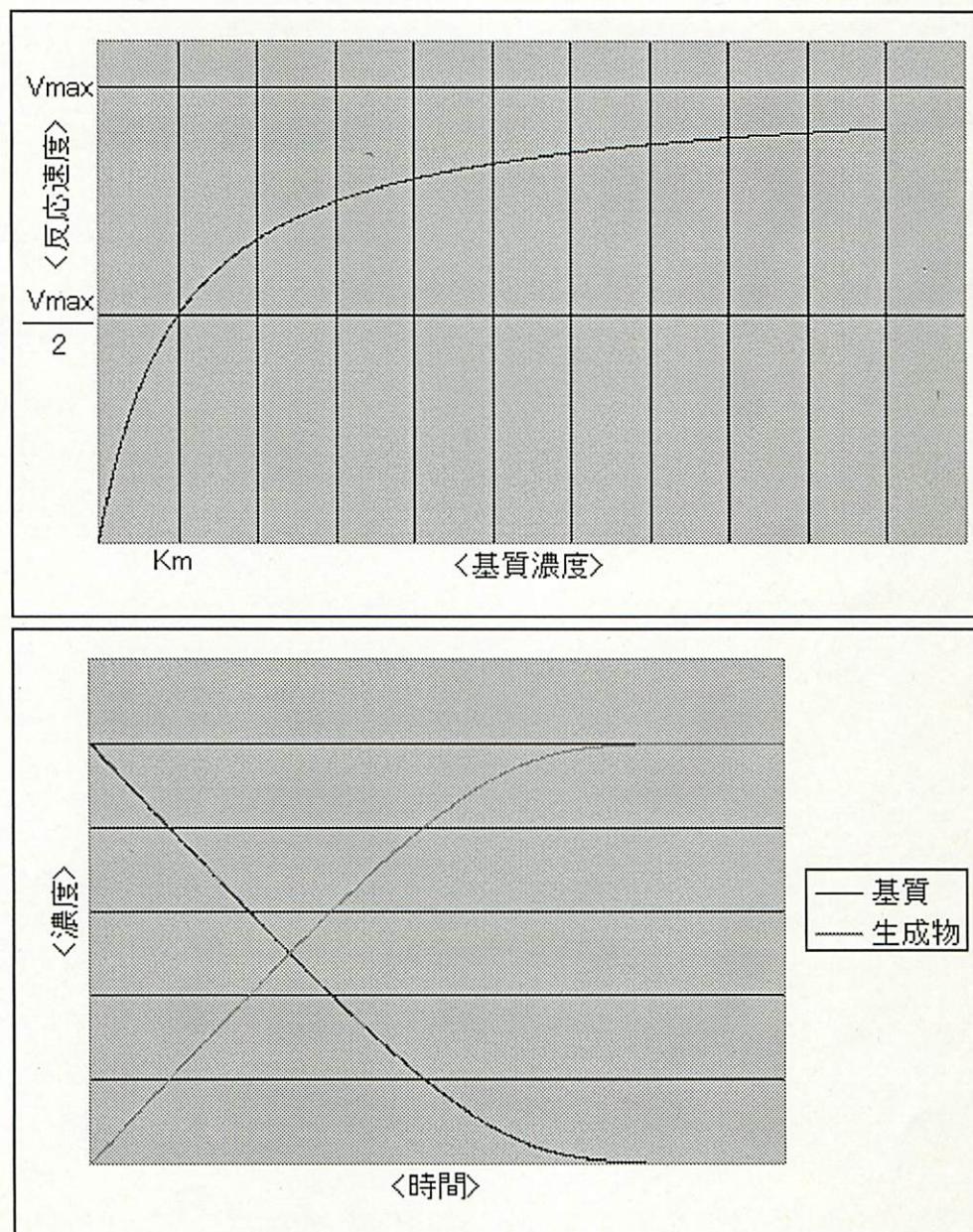
$v = d[P]/dt = k_3[ES] = V_{max}[S]/(K_m + [S])$

ただし、 $V_{max} = k_3[E]t$

$$v = \frac{d[P]}{dt} = \frac{V_{max}[S]}{K_m + [S]} = \frac{V_{max}}{K_m/[S] + 1}$$

として、求められます。 $V_{max}$ は最大速度、 $K_m$ はミカエリス定数と呼ばれます。E-CELLによるモデリングの場合は、通常 $[E]t$ の量が変数となるために、 $V_{max}$ は、 $K_{cat}[E]t$ として表現されます。この場合、 $K_{cat}$ は酵素の回転数と呼ばれ、一つの酵素（もしくは活性部位）当たり単位時間に何回反応するかを表します。

さて、このようにしてもとめられた反応速度式による、反応速度と基質濃度との関係、及び、基質濃度の変化は、以下のようにグラフにプロットすることができます。



この反応の反応速度は、基質濃度が低い場合には、一次反応のように基質濃度に比

例しますが、基質濃度が高くなると反応速度の伸びが遅くなり、 $V_{max}$ を上限として、一定の値に近くなるようになります。また、このグラフからも分かるように、ミカエリス定数は、反応速度が $V_{max}/2$ のときの基質の濃度となります。

※ミカエリスメンテン式による反応のために、E-CELLの標準Reactorとして、MichaelisMentenUniUniReactorが用意されています。

### その他の酵素反応

ミカエリスメンテン式は单基質、单生成物で、不可逆である反応にのみ適用されますが、反応の種類は他にも数多くあります。そして、それぞれについて別々の式を立てる必要があります。

例えば、ミカエリスメンテン式に、逆反応を加えた場合は、反応速度式は以下のようになります。

$$v = \frac{(K_f[E]t[S]/K_m S - K_r[E]t[P]/K_m P)}{(1 + [S]/K_m S + [P]/K_m P)}$$

※逆反応のあるミカエリスメンテン反応のために、E-CELLの標準Reactorとして、MichaelisMentenUniUniReversibleReactorが用意されています。

また、酵素反応の大きな特徴として、阻害、活性があります。その中でも最も重要なものは、フィードバック阻害と呼ばれるものです。これは、代謝系の最終生成物が、その系の入り口付近の反応を阻害することにより、必要以上の物質を生産するのを抑えるものです。阻害の種類には、競合阻害、反競合阻害、混合阻害、アロステリック調節等があり、それぞれについて別々の式が必要となります。

自分で式を立てるのが困難な場合でも、各酵素については詳細に調べられていることもあるので、まず、過去の実験データや、そこから導き出された反応速度式がないかを調べてみるのが良いでしょう。また、比較的標準的な反応については、E-CELLが標準Reactorを用意している場合があります。

\* これらの方針で反応式を得ることができなかった場合でも、基本的に、反応構造と、各素反応の反応速度定数が分かれれば、反応速度式を立てることができます。反応構造から自動的に反応式を立ててくれる「REFERAS」等のソフトウェアがあるので、複雑な反応の速度式を求める場合、このソフトウェアの使用を推奨します。

しかし、反応構造と各定数は判明していないことが多い、また、実験的に調べることも困難なため、酵素反応を定式化することは難しい作業です。特に、各定数についての情報の少なさは、モデリングを行う上での重要な障害の一つとなっています。これらの各定数を実験的に容易に得る方法が開発されるかどうかは、細胞シミュレーション界の将来に大きく関わるでしょう。

代謝系内の反応には、比較的重要なものとそうでないものがあります。反応構造や各定数が判明していない場合でも、重要でない反応については、全体のモデルに与える影響は少ないため、一次式や、ミカエリスメンテン式等の単純な式に近似するという方法があります。

それに比べ、重要な反応についてはより慎重に対応する必要があります。反応の重要度は、代謝制御解析(Metabolism Control Analysis, MCA)等により判断することができます。特に、阻害等が関わるために反応速度が遅くなる反応は律速反応と呼ばれ、その反応を触媒する酵素を律速酵素と呼びますが、これは重要な反応の一つです。各代謝系の速度は、律速反応の速度に大きく依存します。このような重要な反応の場

合は他の式に近似するのは適當ではありません。幸い、重要な酵素については、他のものに比べて、詳しく調査されている場合が多いです。

## 酵素反応以外の反応

細胞内には、酵素反応以外にも数多くの反応が存在します。タンパク質同士の複合、染色体への結合、物質の膜を介した輸送、転写、翻訳、タンパク質の折り畳み、DNA複製等です。E-CELLでは、これらの全ての反応についてモデルを構築することが可能です。

---

Last modified: Tue Nov 28 JST 2000



## E-CELL 生物学の講義 第五回

---

1. 導入
  2. 微分積分とシミュレーションの関係
  3. E-CELLでの計算機構の概略
  4. 積分機構における誤差の低減
  5. PReactor
- 

### 導入

前回は、反応の仕方についてのモデリングの方法を学習しました。物質と反応の関係をルールとして定義し、反応モデルを作ることにより、シミュレーションを行うことができます。

さて、E-CELLはかなり汎用性の高い細胞シミュレーション環境です。そのため、多様な対象に合わせて、複雑なモデルを柔軟に作成することが可能です。そのためには、E-CELL内部でどのような計算が行われているのかを知ることが、大きな助けとなるでしょう。

前回、E-CELL内部での計算手順の簡単な説明をしましたが、今回は、より詳細な理解を進めています。さらに、積分機構による誤差の解消や、微分方程式による時間変化以外の計算方法を用いる場合についてなどについても説明をしていきます。

### 微分積分とシミュレーションの関係

まず、E-CELL内部での計算が数学的にとらえるとどういうことなのかを、非常に簡単に説明します。

E-CELLによるシミュレーションは基本的には力学系のシミュレーションです。

「力学系」とは、次のフェーズの状態がその前の状態から求まる系のことです。次の状態の求め方は、微分方程式等から導かれます。力学系のシミュレーションは、連続的に次のフェーズの状態を計算することによって進められます。この計算は、積分と呼びかえることもできます。

状態は、ベクトルで表されます。ベクトルとは、一次元の配列のことで、もっと簡単に言うと、複数の数字の集まりのことです。E-CELLにおいては、物質の量の集合ということですね。

「微分方程式」は、微分を内部に含む方程式のことです。「微分」とは、ある関数のある変数が微小量変化したときのその関数の変化量を表す関数のことです。

力学系では、もともとの関数が未知であっても、積分の作業によって微分方程式の連続的

な計算を行うことにより、もとの関数の解に近い結果を得ることができます。

## E-CELLでの計算機構の概略

では、実際E-CELLの内部では、この力学系の計算をどのように行っているのでしょうか。そのことを理解するためには、まず、オブジェクト指向のクラスという概念について、知っておくべきことがあります。それは、クラスは変数と関数(メソッド)をその属性として持つことができるということです。

- ① 物質を表すクラスであるSubstanceは、現在の分子数(quantity)とそのステップでの変化量(velocity)を変数として保持しています。
- ② 反応を表すクラスであるReactorは、Substanceのvelocityに値を加算するメソッド(react)を持っています。前回に学習したのは、このreactの設計の仕方についてです。

また、一部のReactorは、reactによって行われる操作が異なり、Substanceのquantityを直接変更します。この先、このようなReactorをReactorと区別してPReactorと表記します。

E-CELL内部では5段階に分けて操作を行っています。clear, react, transit, postern, updateです。操作が行われる順番に説明します。

integrate(積分)

clear

全てのSubstanceのvelocityを0に戻します。

react

全てのReactorのreactを実行します。それにより、Substanceのvelocityに値が加算されます。

transit

*一つの物質の量が複数の式に影響される場合*

全てのSubstanceで、quantityにvelocityを加算します。(場合によっては結果的に減算になります。)

postern

全てのPReactorのreactを実行します。ここまで操作で、そのステップでのSubstanceのquantityが決定します。

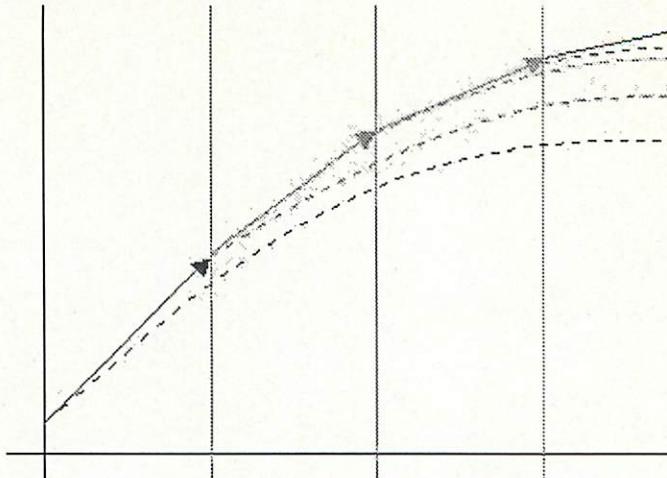
update

GUI等にそのステップでの値を反映させます。

これを1ステップとして、このステップを繰り返すことによってシミュレーションが進んでいきます。

## 積分機構における誤差の低減

上で、「力学系では、もともとの関数が未知であっても、積分の作業によって微分方程式の連続的な計算を行うことにより、もとの関数の解に近い結果を得ることができます。」と書きました。しかし、この積分計算によりある程度の誤差が出る場合があります。そして、シミュレーションではこの計算を非常にたくさん行うので、この誤差は無視できないものとなります。そこで、この誤差を低減させる必要がでてきます。



数値積分による誤差の模式図

本来、黒い点線で表されるべき曲線が、矢印で表されるような線として計算されてしまします。

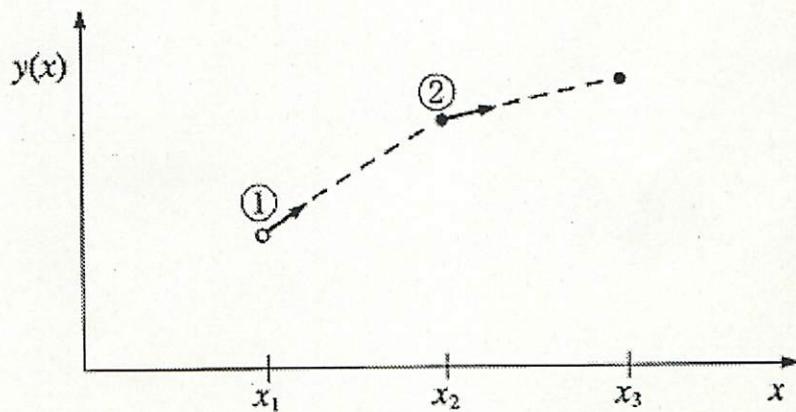
タイムステップサイズを小さくするのが一つの解決方法ではあるのですが、より効率的な数値演算方法があります。それは常微分方程式の数値解法として研究されているもので、様々な方法が存在します。この数値解法は、方法によって様々な特徴があります。誤差は大きいが計算速度が速かったり、その逆だったり、特別な場合において効率が非常に良かってたりします。E-CELLでは各反応系毎に最適な数値解法選ぶことができるマルチステップ一という画期的な方法を使っています。(現状では、開発中のためEuler法と4次のRunge-Kutta法しか使用できませんが)

この数値演算は、実は、前章のreactの段階において行われています。上では分かりやすさを優先して、ただreactを実行していると書きましたが、ここで、より複雑な数値演算を行うことによってvelocityを決定しています。

さて、それぞれの解法について簡単な説明を行います。

### Euler法

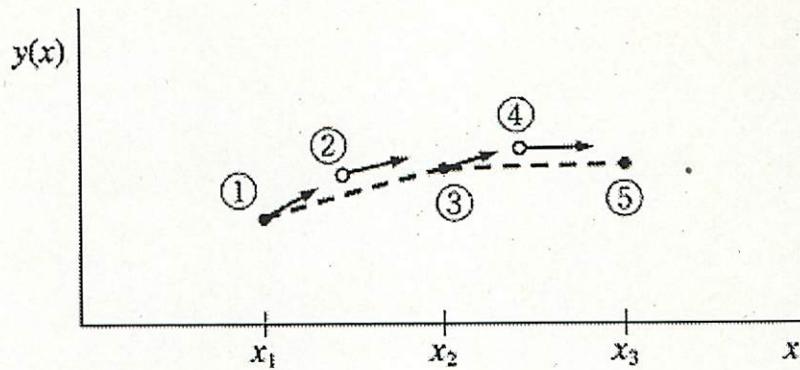
これは、最も単純な方法です。前回に簡単に説明した方法がこの方法にあたります。現在の物質の量から計算された変量(分子数/秒)にステップサイズをかけて、そのまま現在の量に加算することにより次のステップの量を算出します。



Euler法は、誤差が大きいため、実用には向きません。しかし、何らかの理由で誤差の解消を行わない方が良い系の場合はこの方法を用いるのが適しています。

### 2次のRunge-Kutta法(中点法)

誤差を生む大きな原因の一つは、そのステップの始まりの時点での変量(傾き)を、そのステップの間中持続した結果の計算になってしまふからです。2次のRunge-Kutta法では、これを回避するために、そのタイムステップの中点での変量を使用します。

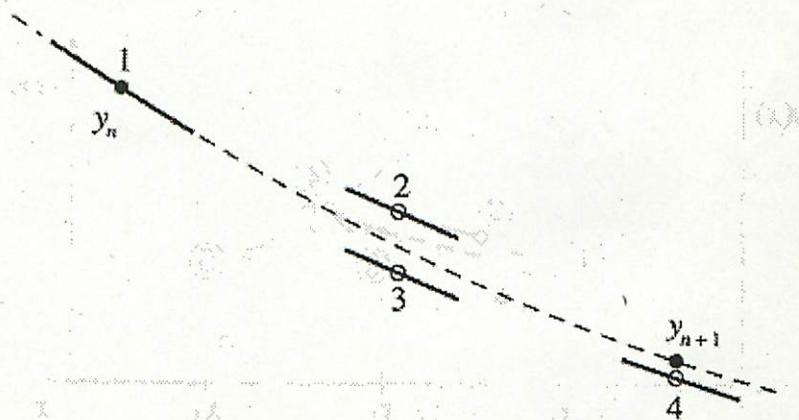


まず、ステップサイズの半分のステップサイズでEuler法と同様の計算を行い、その1/2ステップ後の分子数を求めます。そして、その1/2ステップ進んだ状態での変量を求めてステップサイズをかけ、最初の分子数に加算して、次のステップの量を算出します。

この方法は、Euler法に比べて格段に誤差を解消します。しかし、同様な方法をさらに数手順行うことによってより誤差を少なくする方法もあり、実際にはそちらのほうが有用なため、あまり利用されません。

#### 4次のRunge-Kutta法

この方法は、2次のRunge-Kutta法と同様の作業をもう2手順繰り返します。これによって、より誤差の少ない方法となっています。その分計算の手間が2倍かかるのですが、2次のRunge-Kutta法を、ステップサイズを1/2にして行ったときよりも誤差が少なくなることが経験的に確かめられています。



また、より高次の方に比べても、4次のRunge-Kutta法は優れていることが確かめられています。そのため、この方法が最適である場合がほとんどです。

#### 適応刻み幅制御を取り入れた4次のRunge-Kutta法

適応刻み幅制御とは、タイムステップサイズを可変にすることにより、演算の効率化を図ろうとするものです。変動が大きい時には細かくステップサイズを取ることによって詳細な計算を行いますが、変動が少ない場合にはステップサイズを大きくして、計算を速めることができます。このことによって計算効率は何千倍にも、それ以上にもなる

ことがあります。

4次のRunge-Kutta法に適応刻み幅制御を取り入れる場合は、そのときのステップサイズと、その1/2のステップサイズで計算を行って、それぞれの変量を比較します。それによって、現在のタイムステップでそのまま続けるか、ステップサイズを変更して計算をやり直すかを決めます。また、変更するときは、比較結果からどれくらい増減させるのかを決めます。

これらの方法の他にも、系の挙動を表すヤコビ行列を基に計算するGear法、修正中点法、Richardson補外、Bulirsch-Stoer法などがある。

## PReactor

E-CELL によってシミュレーションを行うためには、シミュレーションモデルに合った Reactor を作る必要があります。

Reactor は、Substance オブジェクトの量を変更することができるオブジェクトです。ユーザーは、Reactor が Substance の量の変化を計算する手順について自由に定義することができます。一方、計算した結果をもとに Substance オブジェクトの量を変更する方法については、今のところ、用意された二種類のメソッドから一つを選んで指定しなくてはなりません。どちらの変更方法を選ぶかによって、Reactor は二種類に分類されます。

### Normal Reactor と Postern Reactor

ひとつは “Normal Reactor” で、積分機構を使って連立微分方程式を解くために使われます。反応速度を計算して、“velocity” メソッドによって Substance オブジェクトの量を変更します。velocity メソッドは、直前のステップにおける Substance オブジェクトの量に、計算結果を差分として与えます。

もうひとつは “Postern(裏口) Reactor” です。Substance の量を計算するために連立微分方程式が適さない場合に使います。次のステップにおける Substance オブジェクトの量を計算して、結果を “SetQuantity” メソッドによって Substance オブジェクトの量に上書きします。作ったリアクターが Postern Reactor である場合、Reactor の名前の最初に “!” をつけ、さらに “PReactor” で終わるようにする、例えば “!ABCPReactor” のような名前をつけることで E-CELL に知らせる約束になっています。

### Postern Reactor は posterior フェイズで実行される

Substance オブジェクトの量が直接上書きされる Postern Reactor は、ひとつのシミュレーションステップの最後に計算されなければなりません。そうしないと、積分の計算をしている途中で Substance オブジェクトの量が上書きされてしまうことになります。次の積分計算はその値をもとに行われる所以、シミュレーション結果が Reactor の呼ばれる順番に依存して変わってしまいます。これではシミュレーションは失敗です。

E-CELL では、全ての積分計算が終わった後で、その値をもとに Postern Reactor による Substance オブジェクトの量を計算、変更するというきまりになっています。Normal Reactor と Postern Reactor は、E-CELL の計算エンジンによって別々に処理されるのです。まず、Normal Reactor による Substance オブジェクトの量の変更が “react” フェイズで実行されます。react フェイズが完了すると、“posterior” フェイズに入り、Postern Reactor による変更が行われます。その後、システムをアップデートします。このへんの下りは、前述の「E-CELL 内部での計算手順」も参照してください。

## ! Postern Reactor 使用上の注意

前述の理由により、Postern Reactor によって操作される Substance オブジェクトは重複しないようにしなければなりません。もし複数の Postern Reactor が重複してある Substance オブジェクトを扱っていると、別々のリアクターがひとつの Substance オブジェクトの量を上書きしあってしまいます。Postern Reactor で表現したい複数の反応が、ある Substance に同時に関係している場合は、それらの反応をまとめて一つの Reactor で表現する必要があります。

Postern Reactor は、具体的にどういう場合に使うの？

Postern Reactor を使うほうが都合がよいのは、Substance の量が離散的に制御される場合です。

- Substance の量が極めて少ない場合。
- Substance の量が、積分によらずに、直接決定される場合。

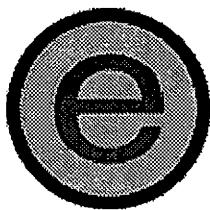
例:

1. リアクターが転写の開始を決定する
2. 細胞分裂によって全てのサブスタンス量が2分の1になる
3. 代数方程式によって反応が計算される

代数方程式による計算は、リアクターが迅速平衡の連立方程式を解く場合や、浸透圧の計算でよく用いられます。

---

Last modified: Tue Dec 5 JST 2000



## E-CELL 生物学の講義 第六回

---

1. 導入
  2. 最適化問題
  3. Rosenbrock法
  4. 修正Powell法
  5. 遺伝的アルゴリズム
  6. パラメータ推定の今後
- 

### 導入

細胞の状態の記述には各物質の濃度や反応速度式の各種パラメータなどの値を入力する必要があります。これらの値は実験で得られたものを論文を検索して入力するのが理想ですが、すべての値が得られるわけではありません。そういうた未知のパラメータに関して、今までには、だいたいこれくらいだろうという、あまり根拠のない値を入力してきました。しかし、より詳細なモデルの記述をするためには、それらの値もしっかりと根拠のあるものにする必要があります。

そこで、今回は、未知のパラメータの値を自動的に推定するための手法、方法、及び、その有効性等について学んでいきます。

また、現在、未知のパラメータの値を自動的に推定するための技術を開発する研究が E-CELL Project内で行われています。その成果は今のところ、ある程度形になってきており、実際にその推定機構を用いたモデリングも行われています。

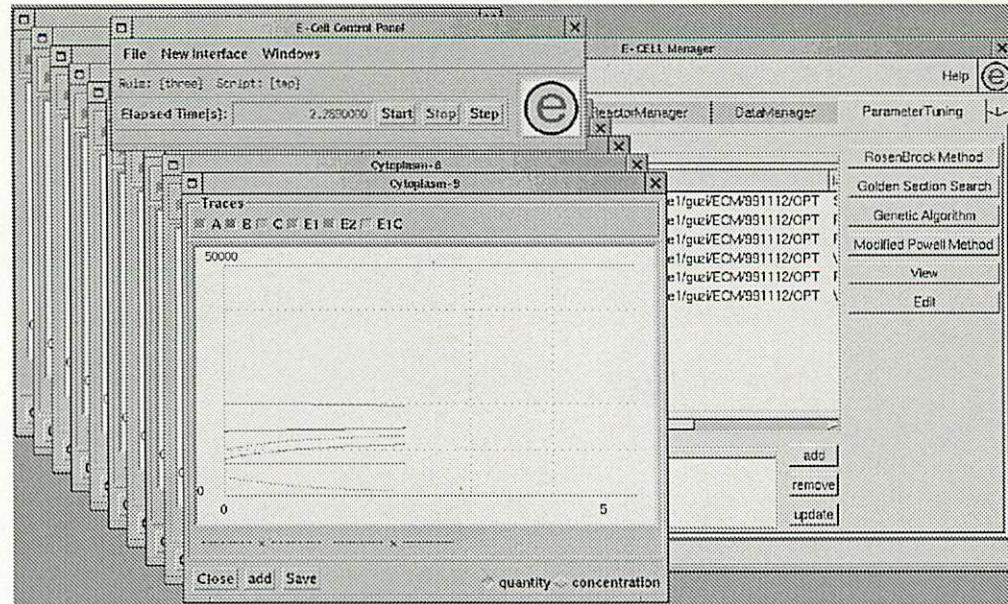


図1:E-CELL Systemを用いたパラメータ推定

## 最適化問題

未知のパラメータを推定することを、未知パラメータを最適化するといいます。一般に、未知パラメータの組み合わせを最適のものにするということを、**最適化問題**といいます。

ひとくちにパラメータが未知だから推定したいといっても、何もヒントがなければ推定のしようがありません。そして、実際に実験が行われている場合には、その実験結果からヒントを得ることができます。

ここでは、実験である物質の値の時間変化がグラフで得られているとします。もし、このデータがあるなら、E-CELLの実行結果がこれと等しくなるようにすればいいのでは?というのがパラメータ推定を行う際のポイントです。前に習ったと思いますが、シミュレーションの利点として、初期値を微妙に変えて何回もくりかえしシミュレータを実行してみるとことができます。この性質を使い、未知変数の値にいろんな値を入力してみて、実験結果に近いものを探すというやり方を用います。

しかし、ただやみくもに適当な値を放り込んで、実行してみればいいわけではありません。その推定値の決め方にもいくつかのルールを適用します。そのルールを最適化手法といいます。最適化手法にはいくつかの種類があって、それをこのあと紹介します。最適化問題を考える際に、一番大事なのが**目的関数(評価関数)**です。これは、ある解が本当の値にどれだけ近いかを数値であらわすためのものです。この関数の値が小さければ理想的な答えに近く、大きければ真の値から離れているということです。この目的関数をどのように計算するかにもいろいろな方法が存在します。現在のE-CELLでは**最小二乗法**というものを使っていきます。

最適化問題とはある目的関数を制約条件のもとで最小にする解を求めることです。上で述べたように、E-CELLではシミュレーションによる計算値を、実験で得られた値に近づけるというアプローチをとるため、目的関数は実験値と計算値の二乗差になります。最適化のためには、この二乗差の合計が最小になるパラメータ(のセット)を探すことになります。これを最

小二乗法といいます。この場合、目的関数は

$$F = \sum_{i=1}^N \sum_{j=1}^n \left( \frac{x_{cal,i,j} - x_{exp,i,j}}{x_{exp,i,j}} \right)^2$$

となります。ここで、 $X_{exp}$ は実験値、 $X_{cal}$ は計算値、 $n$ は測定できる変数の数、 $N$ はサンプリングポイント数です。

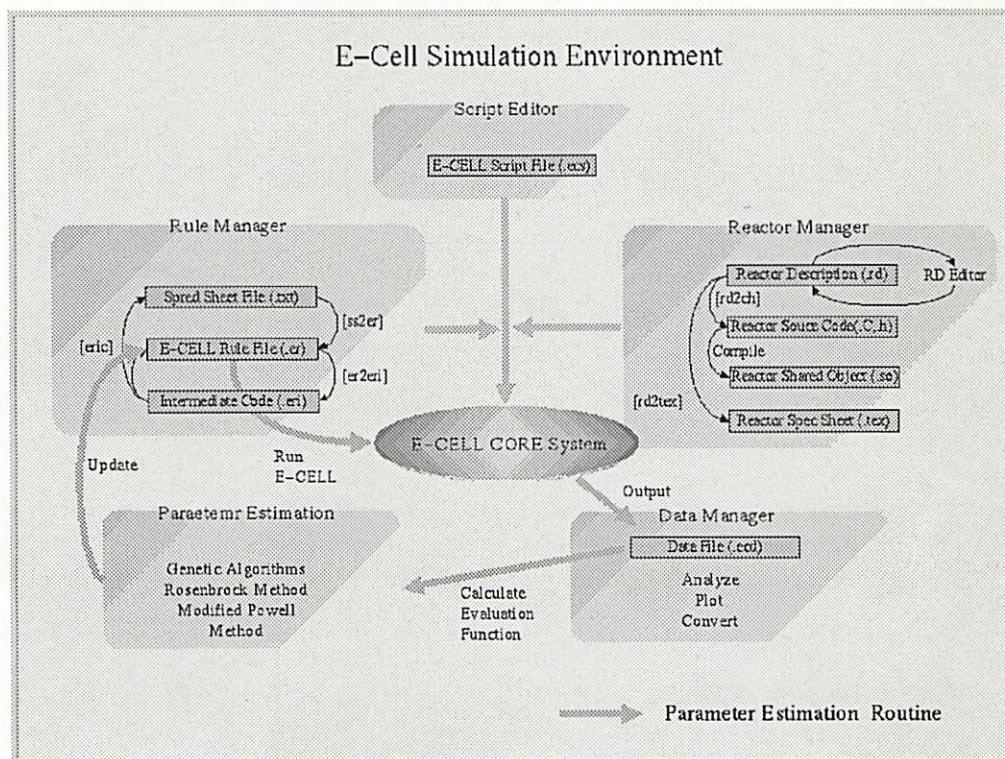


図2:E-CELL Simulation Environmentのパラメータ推定機構(クリックすると図を拡大)

E-CELL Systemを用いたシミュレーション環境の総称をE-CELL Simulation Environmentといいます。パラメータ推定機構はこの一部です。上の図の緑の矢印がパラメータ推定を行う際のプロセスを表現したものです。

最適化手法が推定値を決定、それをルールファイルに代入します。そのルールを用いてE-CELL Systemを実行し、その結果をデータファイルとして保存します。そのデータから評価関数値を計算し、それをもとに最適化手法が次の推定値を計算します。これを決められた終了条件を満たすまで繰り返すことにより、パラメータの推定を行なうのです。

## Rosenbrock法

Rosenbrock法は未知パラメータが1個の場合に用います。この場合1次元空間における探索となります。

Rosenbrock法は別名「成功と失敗のルーチン」といい、ステップ幅を動的に変化させることによって効率的な探索を可能にしています。

グラフでは右方向が正、縦軸は評価関数值です。がある地点から次の地点を進んだとき、そこが前の地点よりも低い(評価関数值が小さい)場合は成功だとし、ステップ幅を $k$ 倍( $k > 1$ )に大きくなります。もし高い場合は谷の底を越えてしまったと解釈し、失敗であったのでそこから逆にステップを戻します。ステップ幅は $-k^l$ ( $l \geq 0$ )この方法を用いれば、最初は徐々にステップ幅を大きくしていって大域的で高速な探索を行い、ある程度目安をつけたら「失敗」を繰り返すことにより最適解に限りなく近づくことが可能です。

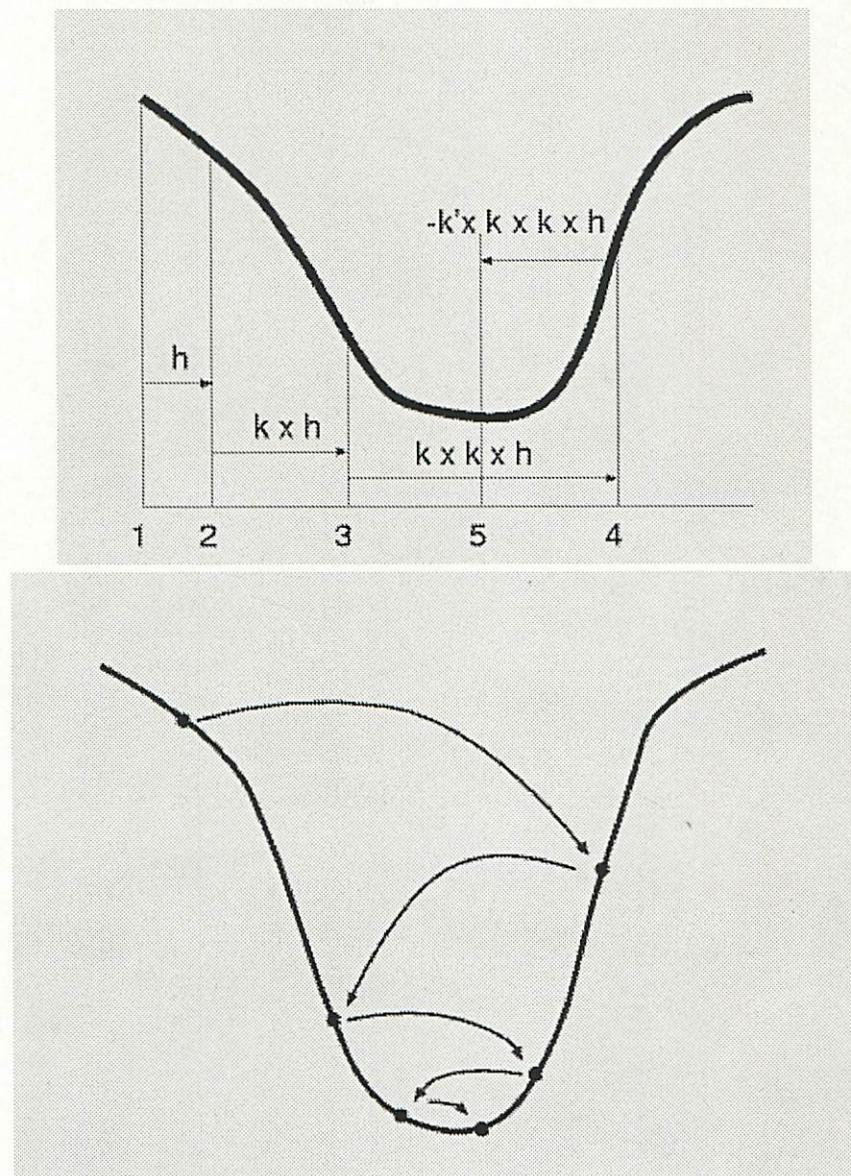


図3:Rosenbrock法 イメージ図

## 修正Powell法

未知変数の数がひとつの場合というのは実際には稀で、ほとんどの場合複数のパラメータの値が分からぬものです。その場合、複数のパラメータを同時に探索しなければなりません。つまり、多次元空間における最適化問題となります。

多変数の最適化手法にもたくさんの種類がありますが、ここでは2つの方法を紹介します。その一つ目が修正Powell法(MP)です。

MPでは推定する解を逐次的に求めていきます。ある値での評価関数を計算し、それが減少する方向に進んできます。ある時点での推定値が決まった際に、次のステップでの値が一意に決まることから、これは決定論的な推定方法であるといえます。

理論的には、この方法でどんどん最適解に近づいていけば、とても精密な解が得られるはずです。実際に、この手法は未知変数の数が小さく、また初期推定が最適解に近い場合はとても有効な手段であるといえます。

しかし、決定論的な最適化手法の欠点として、局所解(Local Minimum)の問題があります。これは目的関数の形状が多峰性の場合におこる問題で、真の最適解を見つけられなくなってしまうという問題です。

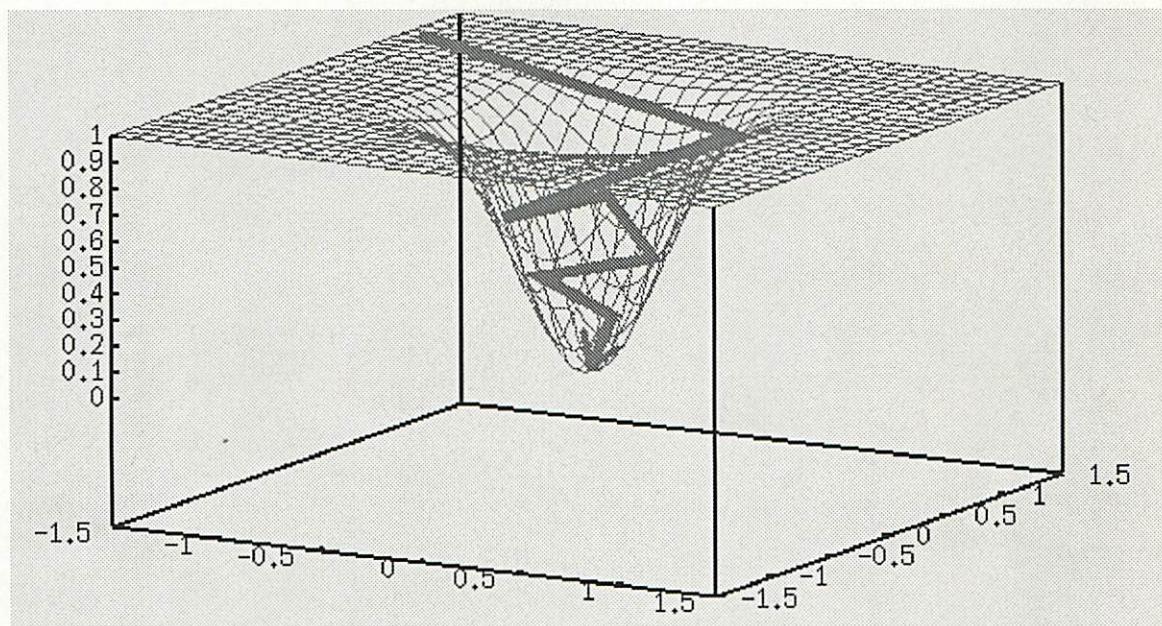


図4:修正Powell法 イメージ図

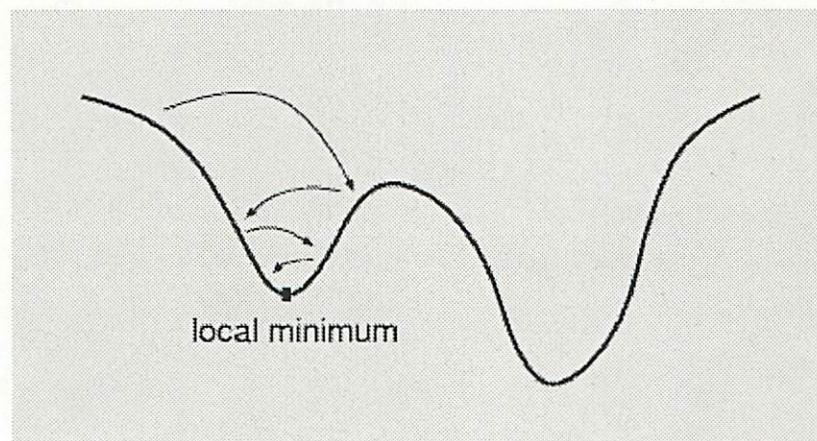


図5:局所解(Local Minimum)

本当の解は右の谷であるのに、アルゴリズムによって探索ポイントが移動する過程で、左の小さな谷のほうが解であると勘違いしてしまい、その谷の底に進んでいってしまうことを、「局所解に陥る」といいます。Rosenbrock法、修正Powell法はともに決定論的な最適化手法です。それはつまり初期推定値がその後の最適化に大きく影響を与えることということです。初期推定が最適解の近傍にある場合は急速に収束し、最適解から遠い場合はいつまでたっても収束しません。よって、これらの手法では初期推定をどのようにとるのかが非常に重

要です。わかりやすく言うと、パラメータの値が分からないにしても、だいたいこれくらいの値かわかっていれば、わりと簡単に本当の値がわかるということです。

## 遺伝的アルゴリズム

局所解の問題を解決し、より大域的な探索を可能にするのが遺伝的アルゴリズム(GA)です。E-CELLで必要とされるパラメータ推定は、未知変数の数が多く、目的関数の形状も多峰性になるのがほとんどです。実際に現在モデリング・グループでは主にこのGAが使われています。

GAは生物の進化の過程にヒントを得た原理に基づいています。遺伝子をもつ仮想的な生物の集団を、環境に適応できるように進化させていくものです。その過程で、「再生」「淘汰」「交叉」「突然変異」といった変化を繰り返していきます。具体的にどのようなアルゴリズムかはこちらを参考にしてください。

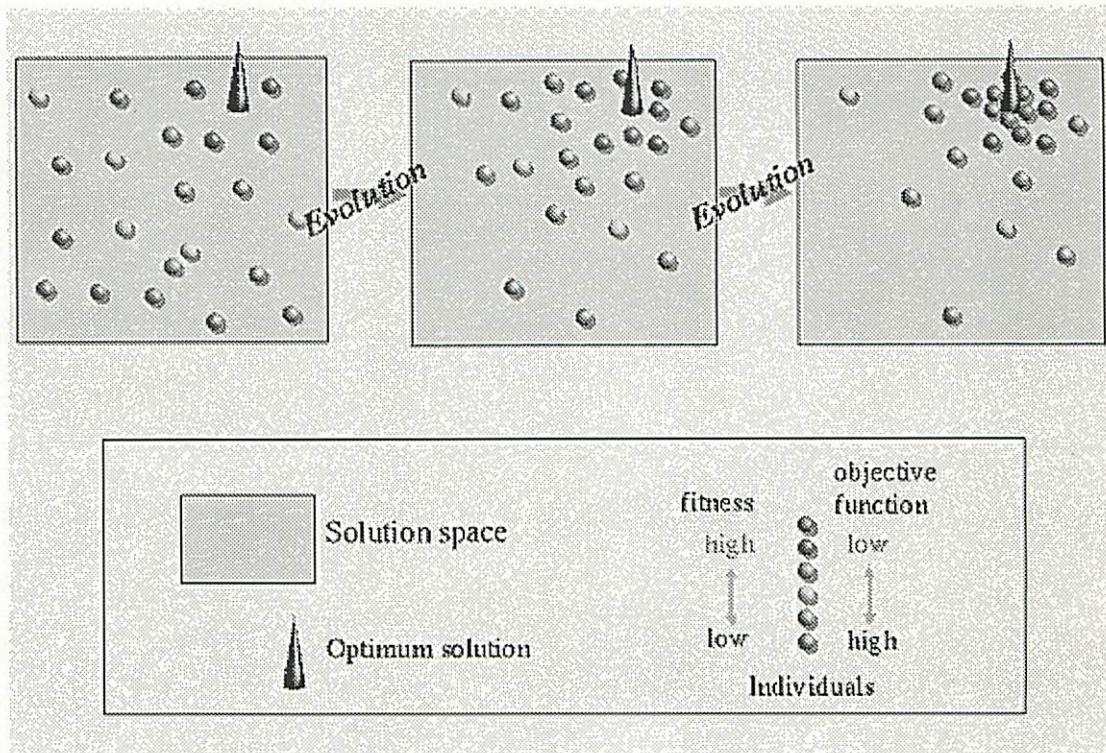


図6:遺伝的アルゴリズムイメージ図(ここをクリック)

GAは進化の過程で確率論的な要素が加わるために、局所解に陥る可能性がありません。また、並列で複数のポイントを探索するために、計算が高速でできるという、利点があります。

## パラメータ推定の今後

現状で、上記の最適化手法が開発されており、それを用いて推定が行われていますが、それですべて問題が解決するわけではありません。この研究はまだまだ途中で、今後解決しなければならない問題点がたくさんあります。

その一番のものが、計算時間の問題です。パラメータ推定を行う場合、推定値を変えて何度もシミュレーションを繰り返すことが必要であるために、通常の何倍もの時間が必要となります。現状の計算速度では満足な結果が得られるために、数週間かかってしまうことがあります。並列処理による高速化などをはかり、より効率的な推定ができるようにしなければなりません。

また、先にのべたのは実験で得られた物質の時間変化のデータがあると仮定しています。しかし、このようなデータがあるとは限りません。その場合、他に方法を考えなければなりません。

それにはいくつかパターンがありますが、定常状態になるようなパラメータを探すというや、系が振動するようなパラメータを探すということが考えられます。このような条件下でパラメータ推定を行う場合は、上記の目的関数の計算方法を他に設定する必要があります。それをどうやるのかは、今後研究を進めていかなければなりません。

---

Last modified: Wed Jan 10 JST 2001



# 第一回 E-CELLを使ってみよう

E-CELLは細胞のシミュレーションを行う環境です。シミュレーションを行うには「ルール」や「リアクター」を用いて表現したモデルを作り、E-CELLに読み込む必要があります。実際は自分でモデルを作成し、シミュレーションを行いますが、まず既存のモデルを用いてE-CELLの操作感を味わいましょう。

## 目標：

E-CELLを用いてのシミュレーションがどのようなものであるかということを体験すること

### SSCモデルで遊ぶ

SSCモデルとはSelf Sustaining Cell(自活細胞)の略で、自活するのに最低限必要な127個の遺伝子を持った仮想の細胞です。この細胞は外部から栄養素を取り込んでATPやリン脂質(細胞膜の部品)を合成し、遺伝子発現によるたんぱく質の合成を行って、自己を維持します。なおSSCモデルは旧バージョンのE-CELLにより構築されており、現行バージョンのE-CELLではシミュレーションを行うことができません。そこで今回は旧バージョンのE-CELLを使用します。今回行う操作に関してはほとんど新バージョンと同じです。最新のE-CELLは次回以降に使用します。

それではE-CELLを用いてSSCモデルのシミュレーションを行いましょう！

1. SSCモデルのコピー
2. E-CELLを起動
3. 基本的な操作

#### 1. SSCモデルのコピー

まずはSSCモデルをWebからダウンロードします。<http://www.e-cell.org/model/index.html>にある”An executable binary file of Self-Sustaining Cell Model and its ancillary files (\*.tar.gz) ”をクリックし、自分のディレクトリにコピーします。次にダウンロードしてきたファイルを解凍します。

```
%tar zxf SSC.tar.gz
```

SSCというディレクトリができたことを確認し、そのディレクトリの中に移動します。次の5個のファイルが存在することを確認して下さい。

SSC.cgi	after50s.cs	default.ecs
ecell	ssc.cs	

これら5個のファイルが存在していれば、SSCモデルのダウンロード、解凍作業は成功です。

## 2. E-CELLを起動

E-CELLの起動ですが、これはいたって簡単です。

**%./eCell**

このようにキー入力するだけです。

## 3. 基本的な操作

- E-CELLを実行する

E-CELLを起動すると、E-CELL Control PanelとMessages Windowが現れます。E-CELL Control PanelはE-CELLを操作するためのMain Windowです(図1)。Messages WindowはE-CELLからのエラー等のメッセージを表示するためのもので、シミュレーションには特に影響を与えることはありません。

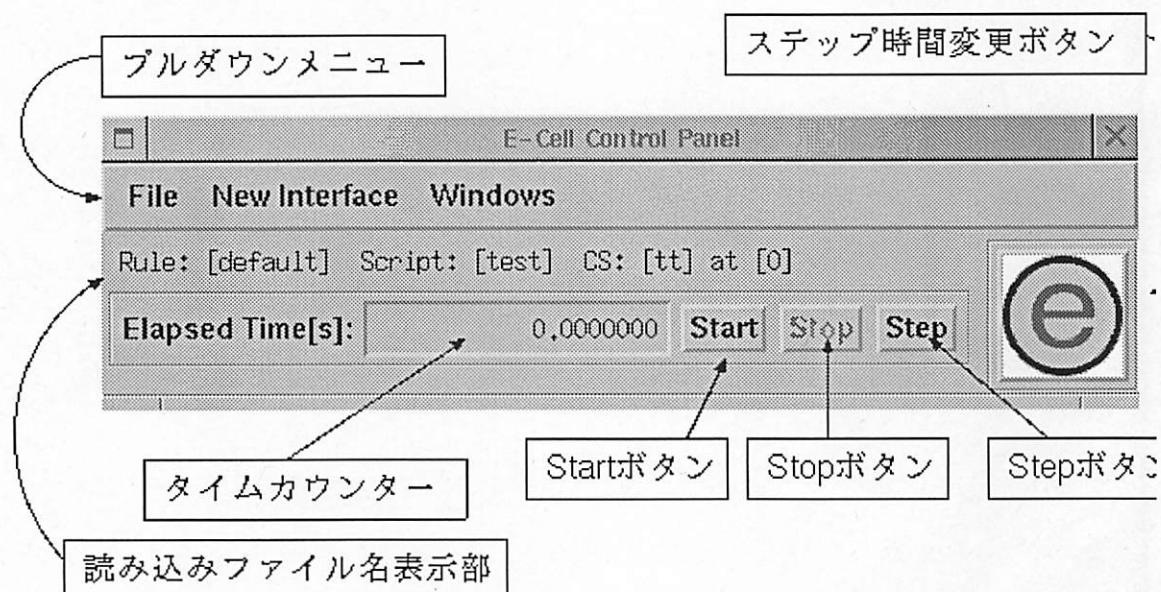


図.1 Control Panel

### ■ Ruleを読み込む(Install Rule)

Ruleファイルは、E-CELLがシミュレーションを行うのに必要な情報が記述されたデータファイルです。このControl Panelにある「File」という欄をクリックします。すると「Install Rule」と「Quit」が選択できる状態になっています。ここでまず「Install Rule」ボタンを選んでRule fileを読み込みます。

### ■ Cell Stateを読み込む(Load Cell State)

ある時点におけるすべての物質の量をとったファイルを「Cell State」ファイルといい、「.cs」という拡張子で示します。今回はRuleファイルにかかれているデータより良いものが「Cell State」ファイルとして存在しているので、そのファイルを使いましょう。方法は次の通りです。Control Panelの「File」をクリックすると、「Load Setup File」「Load Cell State」「Save Cell State」「Quit」の4つが選択可能になっています。ここでまず「Load Cell State」を選択します。すると「Load Cell State\_popup」というWindowが現れます。このWindowの「Files」の欄にある「ssc.cs」というファイルを選択し、下にある「OK」ボタンをクリックします。

- Setup Fileを読み込む (Load Setup File)

その後、Control Panelに戻り、「File」を押し、その中の「Load Setup File」を選択します。すると今度は「Load Setup File\_popup」windowが現れます。このFilesの欄にある「default.ecs」を選択し、「OK」をクリックします。するとびっくりするくらい多くのwindowが現れます。これらがトレーサーと呼ばれ、物質などの量的変化を表示するものです(図2、縦軸は物質の量、横軸は時間)。

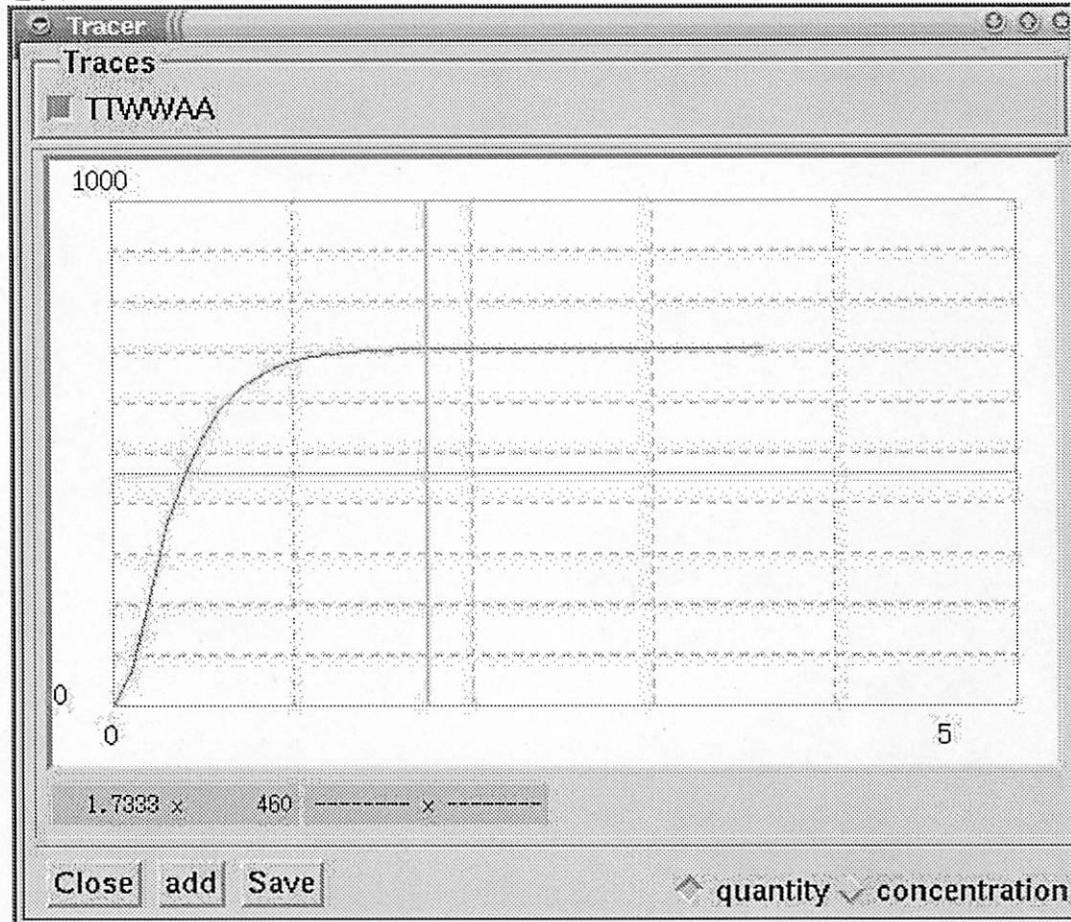


図.2 Tracer Window

ここでControl Panelにある「Run」ボタンを押しましょう。シミュレーションが開始され、トレーサー上にグラフが描かれていきます。

- タイムステップ

タイムステップを変更してみましょう。まずControl Panelの「Stop」ボタンでシミュレーションをとめます。そしてE-CELLマークをクリックします。次に旧バージョンではスケールバーを左右に動かしてください(図3)。新しいバージョンでは入力欄に直接数字を入力します(図4)。タイムステップの変更はシミュレーションの積分時間間隔を変更することを意味します。積分時間間隔を変更することでシミュレーションを短時間で行うこともできます。だったらいつも間隔は最大にしておけばいいじゃないかって思う人がいるかも知れませんが、この積分間隔を大きくするとシミュレーション自体の精度が低くなってしまいます。反対に間隔を小さくすると精度は高くなりますが、シミュレーションに時間を多く費やしてしまい、研究が進みません。つまり自分が行いたいシミュレーションにとってどれくらいの積分間隔が最適かを見つける必要があるのです。

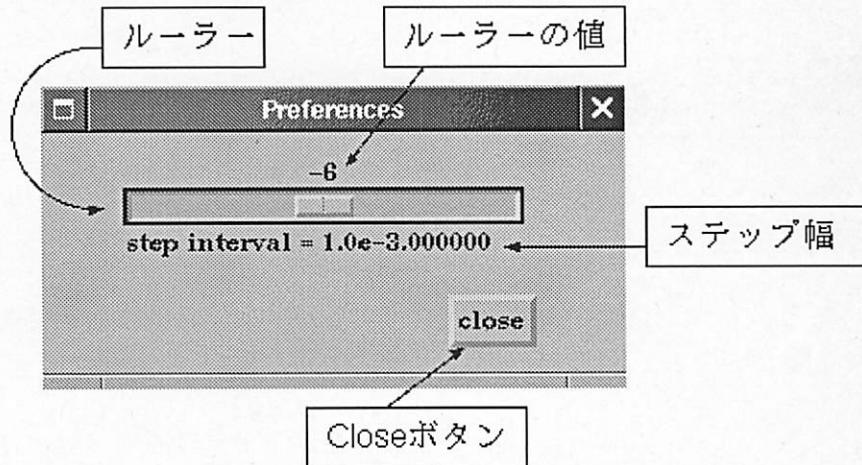


図.3 Preference(旧)

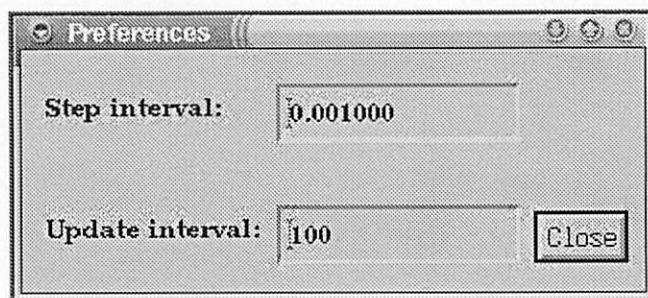


図.4 Preference(新)

タイムステップを変更したら、シミュレーションを再開して、どのように変化しているか観察してみましょう。

#### ○ グラフの拡大

またいったんシミュレーションをとめて、今度はグラフを詳しく調べてみましょう。グラフ上で左クリックをするとクリックした点を交点とする青い十字が表示されます。この青い十字は、複数のTracerWindowが表示されている際は、全て同一の座標に表示されます。この青い十字を消したい時はグラフ表示部外をクリックします。次に、グラフのどこかに中ボタンのドラッグにより矩形を描いてみましょう。その範囲を拡大して表示できます(図5)。この拡大した状態をキャンセルするには右クリックします。

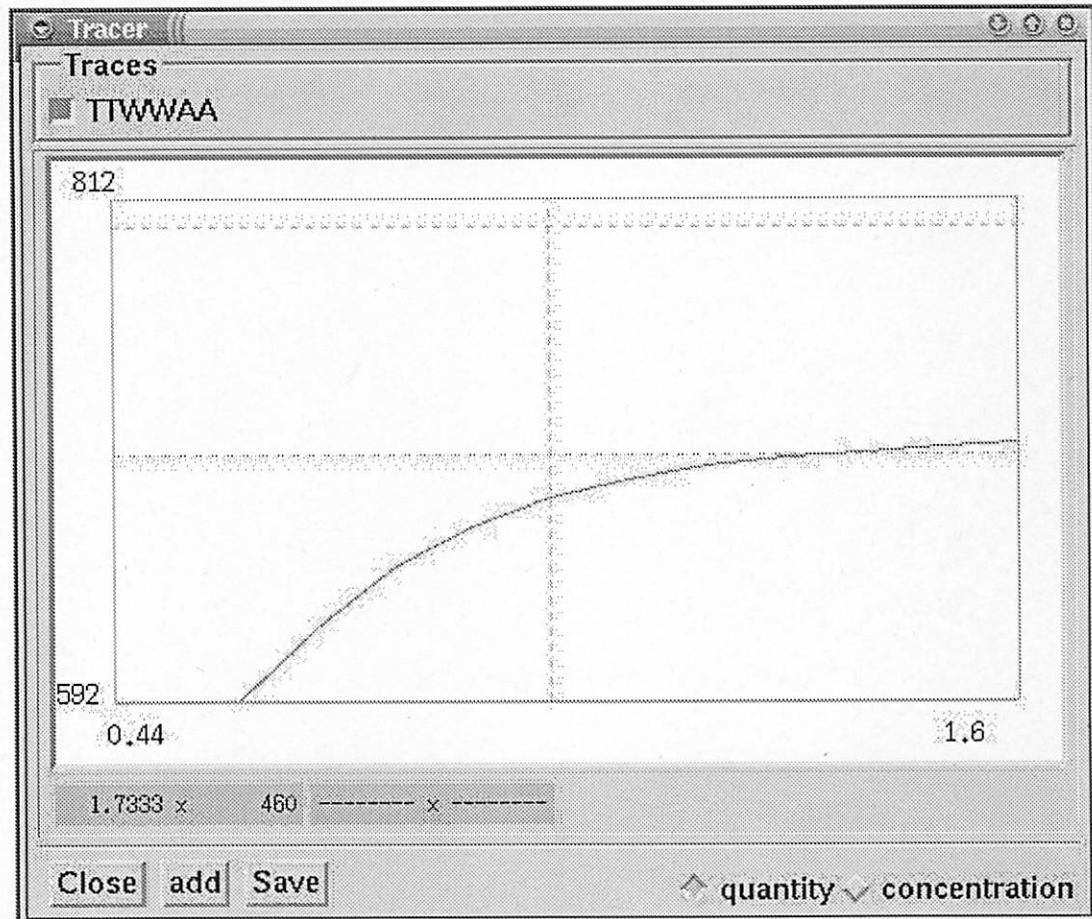


図.5 Tracer Window(拡大)

- 新たなトレーサーを作る

シミュレーションをとめて、新しくトレーサーを作つてみましょ。Control Panelの「New Interface」から「Tracer」を選択します。

- トレーサーにSubstanceを加える

作成したトレーサーにTracer Window上の「add」ボタンをクリックすると新たなEntrySelectorが表示されます(図6)。「System」から「CYTOPLASM」を選択し、その後「Substance」に表示されたものの中から適当なものを選択します。そして「OK」をクリックすると、Tracer Windowに選択したSubstanceが加えられます。

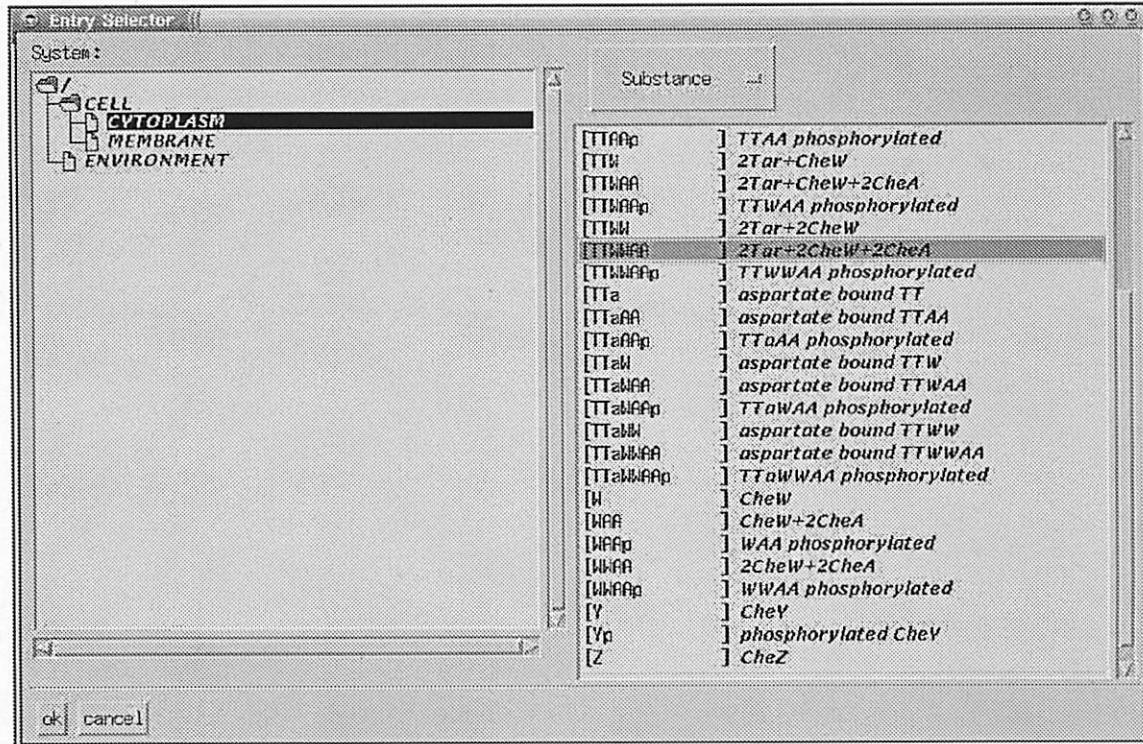


図.6 EntrySelector

- Substance Windowを表示する

Tracerの上部に物質名が表示されています。ここから1つ選んで物質名を右クリックし、Substance Windowを表示させましょう(図7)。ここで現時点での物質の量をかえることができます。選択した物質の量をかえてシミュレーションを再開しましょう。

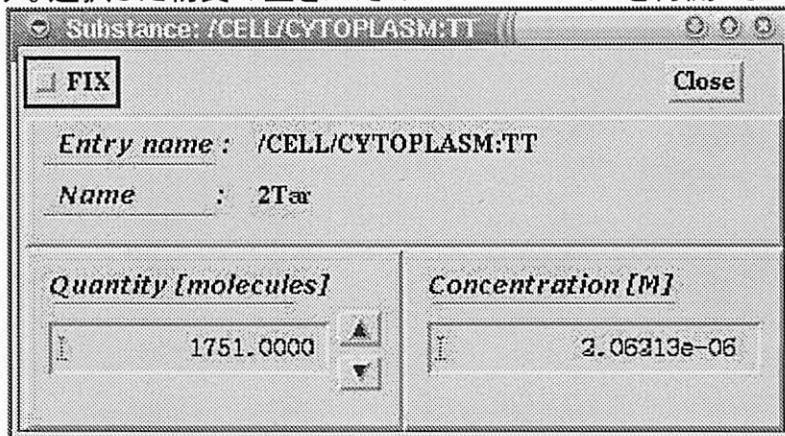


図.7 Substance Window

- トレーサーを閉じる

TracerWindow上の「close」ボタンをクリックします。Windowの右上にある「×」ボタンを押すとすべてのWindowが閉じてしまうので注意してください。

- Cell Stateを保存する

Cell Stateを保存しましょう。まずE-CELL Control Panelの「File」にある「Save Cell State」を選択します。すると「Save Cell State\_popup」windowが表示されます。「Files」欄に、現在そのディレクトリにある「.cs」ファイルが表示されます。その中から選択する

とそのファイルは上書きされます。新たなファイルを指定するときは「.cs」で終わるファイル名を記入します。

- E-CELLを終了

E-CELL Control Panelの「File」にある「Quit」を選択します。

以上で第一回は終了です。

今回のチュートリアルに関する感想、質問等は [e-cell-tutorial@eg.e-cell.org](mailto:e-cell-tutorial@eg.e-cell.org) にお願いします。

---

*Motohiro Yoneda*

---



## 第二回 Toy Pathway を作る ~その1~

---

目標：

パスウェイマップの作図ができるようになること

前回はSSCモデルで遊ぶために最新ではないE-CELLを用いてE-CELLの基本的な操作を学んだと思います。今回からはE-CELLの最新バージョンを用いてシミュレーションするのに必要なファイルの作成について学んでいきます。

そして今回は目的のシミュレーションを E-CELLで実行するための情報を記述するデータファイルである「Ruleファイル」を紹介し、簡単なモデルの図を作成します。

1. Ruleファイルの例解
  2. ToyPathwayを作る～その1～
- 

### 1. Ruleファイルの例解

Ruleファイルは次のようなものです。実際に作成していけば、わかると思います。だからここではruleファイルがどのようなものであるのかというイメージをつかんでください。

```

# this is ecell rule file for ECS-v08
# converted by 0.9.0
include(qty.er)                                必要なファイルをインポート

system Cell(/:CELL,"The cell")                細胞を定義
{
    Stepper SlaveStepper;
}

system Environment(/:ENVIRONMENT,"The culture medium") 環境を定義
{
    Stepper SlaveStepper;
    VolumeIndex /ENVIRONMENT:VOLUME;
}

system Cytoplasm(/CELL:CYTOPLASM,"The cytoplasm") 細胞質を定義
{
    Stepper SlaveStepper;
    VolumeIndex /CELL/CYTOPLASM:VOLUME;
}

system Membrane(/CELL:MEMBRANE,"The membrane") 細胞膜を定義
{
    Stepper SlaveStepper;
    Inside /CELL:CYTOPLASM;
    Outside /:ENVIRONMENT;
}

SETVOLUME(/ENVIRONMENT,1E-12)                  体積を定義
SETVOLUME(/CELL/CYTOPLASM,1.41E-15)

```

図.1

```

reactor ConstantParameterReactor(/ENVIRONMENT:VOLUME,"Volume index"
{
    Value 1E-12;                                環境の体積値
    InitialActivity 1E-12;
}

reactor ConstantParameterReactor(/CELL/CYTOPLASM:VOLUME,"Volume index"
{
    Value 1.41E-15;                            細胞質の体積値
    InitialActivity 1.41E-15;
}

```

図.2

```

#-----#
#          物質の定義          #
#-----#
標記          物質のID   名前          定義する物質の分子数
substance /CELL/CYTOPLASM:TT "Receptor_T_dimer" 4250;
substance /CELL/CYTOPLASM:a "aspartate" 4250;
substance /CELL/CYTOPLASM:TTa "Receptor_and_attractant" 4250;
substance /CELL/CYTOPLASM:TTWAA "TTWAA" 100;
substance /CELL/CYTOPLASM:TTWAAp "phosphorylated TTWAA" 2240;

#-----#
#          反応の定義          #
#-----#
リアクターID          反応のID
reactor Chemo_RapidEquilibriumReactor (/CELL/CYTOPLASM:CF1, "TTa")
{
    リアクターにわたす引数
    Keq 1e6;
    Substrate /CELL/CYTOPLASM:TT 1 1; 反応基質の指定
    Substrate /CELL/CYTOPLASM:a 1 1;
    Product /CELL/CYTOPLASM:TTa 1 1; 生成物の指定
}

reactor MichaelisUniUniReactor (/CELL/CYTOPLASM:PHO_2, "TWAAstmAAp")
{
    KmS 3e-4;
    KcF 5.8e4;

    Substrate /CELL/CYTOPLASM:TTWAA 1 1;
    Product /CELL/CYTOPLASM:TTWAAp 1 1;
}

```

図.3

## 2. ToyPathwayを作る～その1～

### ◦ パスウェイマップの作図

Toy Pathwayとは非常に簡単なPathwayのことで、生命体の複雑さに対してあまりにも単純な系であるため、まるでおもちゃのように思えることから名付けられました。

モデル作りの第一段階はシミュレーションを行う系の各物質がどのように相互に関わっているか、ということを一目でわかるように図にすることです。これをパスウェイマップといいます。

作図方法は次のようになります。物質AがCという触媒を介して物質Bに変化するということを図にすると図4になります。中央の長方形はリアクターです。リアクターというのはE-CELLでは反応を表します。

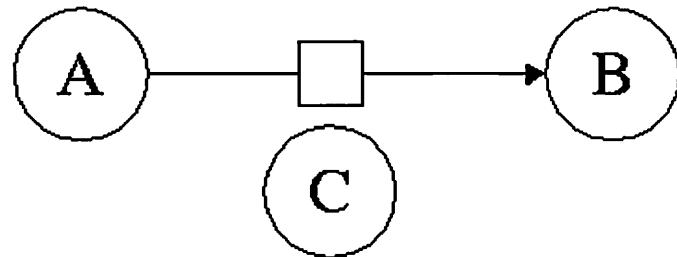


図.4

物質Aと物質Bが結合して物質ABに変化するという現象は図5のように表します。

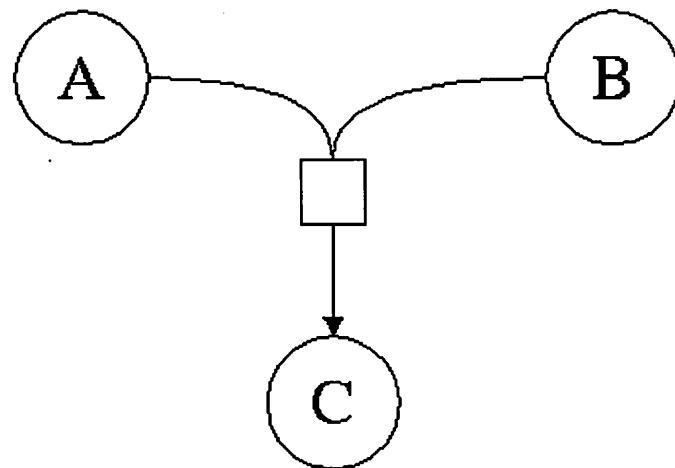


図.5

図5と図6の違いは矢印が一方向か両方向かということです。図6では、図5のように物質Aと物質Bが結合して物質ABに変化する、ということだけでなく反対に物質ABが物質Aと物質Bという2つの物質に変化するという反応も起こり得ることを表現しています。

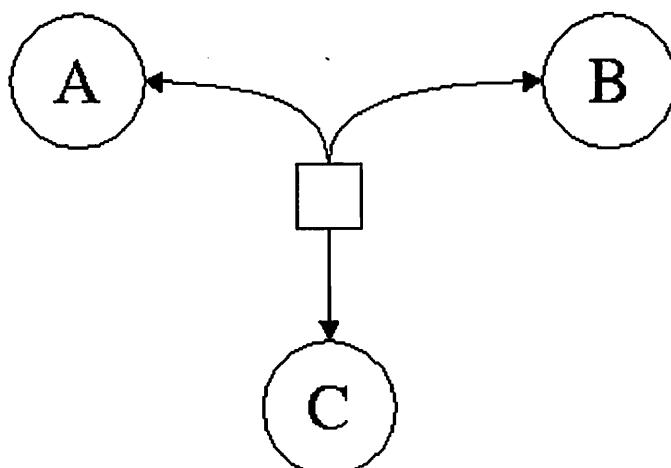


図.6

それではToy Pathwayとして次のものを作図してみましょう。

物質Aが触媒E1を介して物質Bに、物質Bが触媒E2を介して物質Cに変化します。物質Cは触媒E3を介して物質Dに変化します。そして触媒である物質E1と物質Cが結合して物質E1Cが合成されます。また物質E1Cが逆に物質E1と物質Cとに分解される反応もあります。これがもっとも簡単なフィードバック系です。

- スプレッドシート(Star Office、Gnumericなどの)を使えるようになる。

パスウェイマップが完成したら、次にそれを参考にルールを書きます。ルールを書く手段として

1. スプレッドシートに記述してE-CELL Ruleファイル(erファイル)に変換する
2. erファイルを直接書く

という2つの方法があります。そのため、スプレッドシートを使えるようになります。

スプレッドシートとはEXCELなど表計算をするソフトウェアのことです。E-CELLでは一般にStar Officeのスプレッドシートを使用しています。ですからEXCELを使ったことがあれば大丈夫です。

以上で第二回は終了です。

今回のチュートリアルに関する感想、質問等は [ecell-tutorial@eg.e-cell.org](mailto:ecell-tutorial@eg.e-cell.org) にお願いします。

*moto*  
t97075my@sfc.keio.ac.jp



## 第三回 Toy Pathway を作る ~その2~

**目標:**

Toy Pathwayを完成させる。

前回はパスウェイマップを作成しました。今回はそれを元にスプレッドシートを用いてルールを書きます。そしてecellで実際にシミュレーションを行います。最後にシミュレーション結果のデータをセーブし、gnuplotなどのグラフ作成ツールを使ってプロットするというところまで行います。多少ハードですが、がんばりましょう！

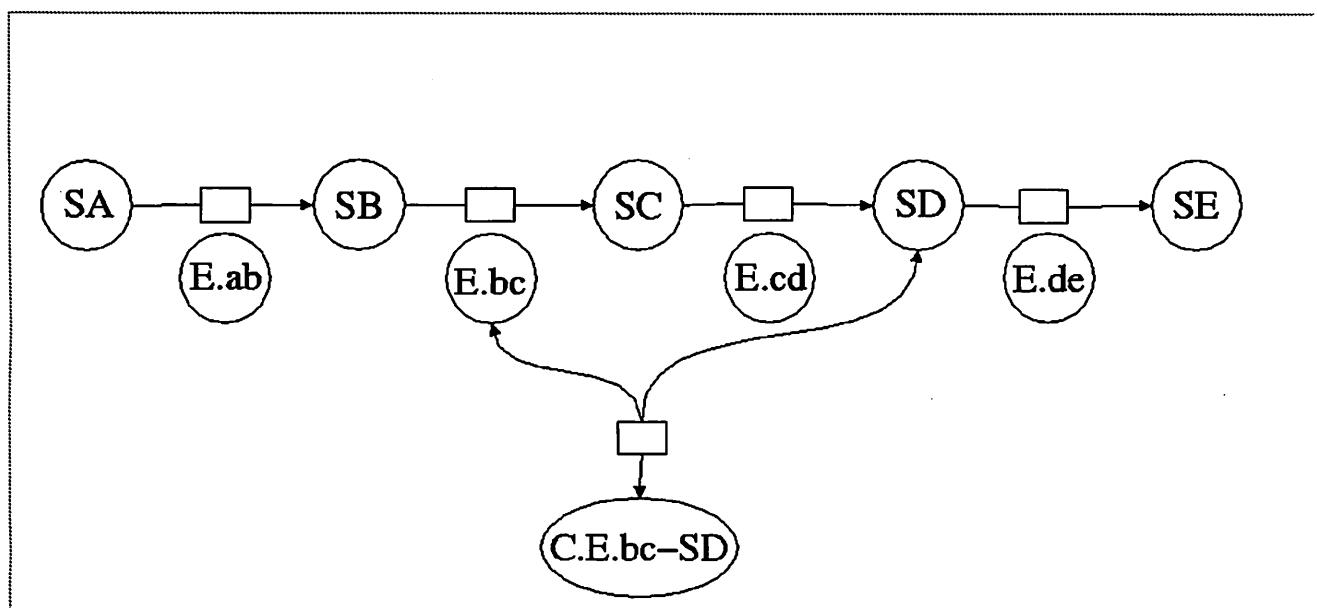


図.1

1. ルールを書く
2. ルールの変換を行う
3. ecsファイルを書く
4. Saveしたデータを出力、プロットする

1. ルールを書く

- Star Officeを起動し、Spreadsheetを用いてルールを書きましょう。

まず tane.sdc をコピーします。これをテンプレートにしてルールを作成します。

スプレッドシートには下の表のような内容を記入していきます。

Type	path	ID	Name	QTY
Substance	/CELL/CYTOPLASM	SA	Substance A	1000
				Fix
Substance	/CELL/CYTOPLASM	SB	Substance B	0
Substance	/CELL/CYTOPLASM	SC	Substance C	0

この表は物質の定義です。Typeには「System」「Substance」「Reactor」の3つがあり、その行にこの3つのうちのどれを記入しているかを示します。ここではSubstanceです。PathはSubstanceのある場所です。IDは指定したSubstanceのIDで、省略することはできません。Nameは指定したSubstanceの名称のことです。QTYはSubstanceの個数(分子数)の初期値(Concとは両立しない)で、ConcはSubstanceの濃度M(mol/l)の初期値(Qtyとは両立しない)のことです。

上の表中のSubstance Aは図1のSAのことです。

課題：

Substance D、Substance Eについては考えて記入してください。QTYの値はともに0です。

Type	path	ID	Name	CONC
Substance	/CELL/CYTOPLASM	E.bc	Enzyme B	0.02
Substance	/CELL/CYTOPLASM	E.cd	Enzyme C	0.01
Substance	/CELL/CYTOPLASM	C.E.bc-SD	Complex of E.bc and SD	0

ここは酵素についての項目です。

課題：

Enzyme A(0.83、Fix)、Enzyme D(0.01)について考えて記入しましょう。

以上は物質についての定義でしたが、下の表は反応を定義しています。

Type	Class	path	ID	Name	S_ID	
Reactor	MichaelisUniUniReactor	/CELL/CYTOPLASM	E.ab-0	A->B	SA	/C
Reactor	MichaelisUniUniReactor	/CELL/CYTOPLASM	E.bc-0	B->C	SB	/C
Reactor	MichaelisUniUniReactor	/CELL/CYTOPLASM	E.cd-0	C->D	SC	/C

ClassはReactorの名前です。S\_IDは基質のID、S\_Pathはその場所、P\_IDは生成物のID、P\_Pathはその場所を指しています。C\_IDは反応触媒(Catalyst)のID、C\_Pathは

Catalystの存在する場所です。Arg\_tag列ではReactorで使用する定数名、つまり反応速度式が使用する定数を入力する。この定数名は、Reactorごとに決まっています。Arg\_coeffには、Arg\_tagに対応する定数の値が入ります。

上の表の3つのReactorは1つの物質が別の1つの生成物を作るという反応だから S\_ID、S\_Path、P\_ID、P\_Pathはそれぞれ1つずつです。

課題：

Reactor の D→Eの反応について記入しましょう。ただしKmSは0.1、KeFは1です。

課題：

それができたら次は Ebc+SD<→EbcSDについて記入しましょう。

これまでのReactorにはMichaelisUniUniReactorを使用しましたが、この反応では RapidEquilibriumReactorというものを使用します。EbcとSDが結合してEbcSDが合成されます。またEbcSDが逆にEbcとSDとに分解される反応もあるということをあらわすようにします。つまり(S\_ID、S\_Path)または(P\_ID、P\_Path)を2個にするということです。ちなみにIDは!EQ-Ebc-SD、Keqは100とします。!EQ-Ebc-SDの「！」は代数式の Reactorであることを示します。

終わったらセーブします。「File」から「Save As」を選び、File Typeを「StarCalc 5.0」とし、適当なファイル名(拡張子をつけなくて大丈夫です)を入力し、「Save」を選択します。この「StarCalc 5.0」というFile Typeでセーブしておかないと、もう一度セーブしておいたファイルの内容を変更したいと思ったときに、スプレッドシートとして開くのが難しくなる可能性があります。

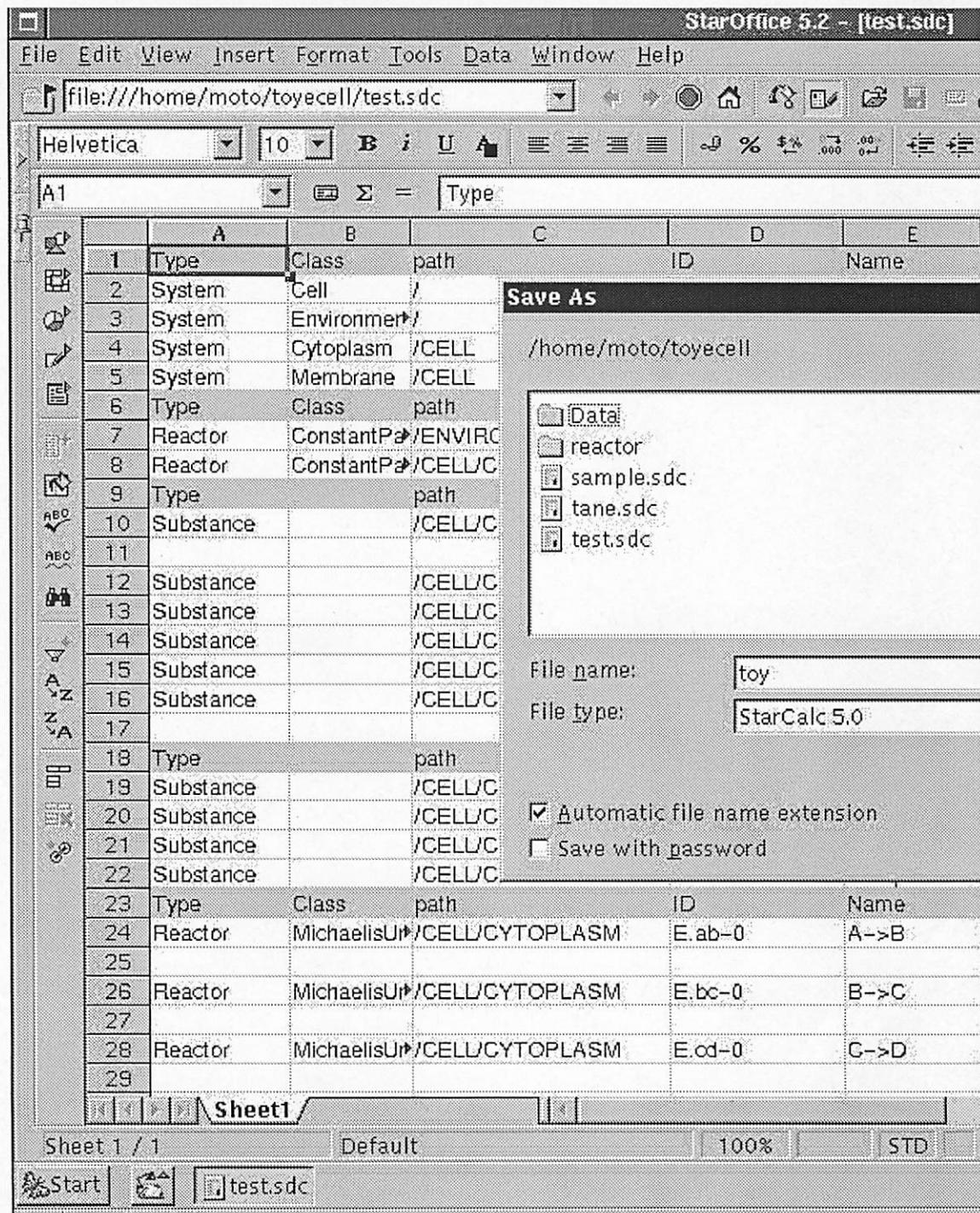


図.2 スプレッドシートのセーブ画面

## 2. ルールの変換を行う

ルールの変換とはこのスプレッドシートを、E-CELLで読める形式に変換することです。

- 「qty.er」というファイルをコピーします。  
/usr/local/share/e-cell/1.0/er/にあります。

```
% cp /usr/local/share/e-cell/1.0/er/qty.er .
```

- スプレッドシート形式のファイルをテキスト形式のファイルに変換します。「File」から「Save As」を選び、File Typeを今度は「Text-txt-csv(StarOffice Calc)」を選択します。するとファイル名の拡張子が自動的に「txt」に変わります。これで「Save」をし、「OK」を

選択します。

- テキスト形式に変換したファイルをE-CELLに読み込めるファイル形式に変換します。  
これはコマンドラインから行います。

```
% ss2er < toy.txt > toy.er
```

```
% er2eri < toy.er > toy.eri
```

汎用のスプレッドシートソフトウェアによるルールファイルの作成は、視覚的に理解しやすく、比較的簡単に記述できることが利点である。一方、erファイルはテキストファイルとして記述されるため、マクロを使って簡単に操作できる。これらはいずれも、最終的にE-CELL Systemに読み込まれる形である eriファイルに変換される(スプレッドシートの場合は一度、erファイルに変換されてからeriファイルに変換される)。

### 3. ecsファイルを書く

- Reactorをコピーする

スプレッドシートで記述したreactorの反応を起させるには実行ファイル(SOファイル)が必要です。そのためにはまず /usr/lib/e-cell/1.0/dm/so/reactor にあるreactorをディレクトリごとコピーします。

```
% cp -rd /usr/lib/e-cell/1.0/dm/so/reactor .
```

- E-CELLを使うここまでできたら第一回の復習も兼ねて1度E-CELLを起動させてシミュレーションしてみましょう。

```
% ecell -r ./reactor
```

-r ./reactor でReactorの場所を指定し、Reactorを読み込みます。

- ecsファイルを書きます。このファイルでシミュレーションを行う際のいろいろな設定ができます。以下が簡単な説明です。

```
ReactorPath ./reactor
LoadRule toy.eri

NewInterface Tracer A B C
AddTrace Substance:/CELL/CYTOPLASM:SA
AddTrace Substance:/CELL/CYTOPLASM:SB
AddTrace Substance:/CELL/CYTOPLASM:SC
SaveAt 500

UpdateInterval 100

Run 501
SaveCellState sample-after500s.cs
Stop
```

図.3 ecsファイルの例

- 1行目:必要なReactorのあるディレクトリを指定します
- 2行目:読み込みたいルールファイル(eriファイル)を指定します。
- 4行目～8行目:表示させたいトレーサはこのように書きます。
  - 4行目～7行目:AddTraceはTracer内でのみ宣言される。引数にかかれたSubstance,もしくはReactorをTracerの表示に加える。引数は、IDだけでなくそれが存在するシステムとTypeを含めて示す必要があります。

- 8行目: このトレーサ上に表示させた物質を500秒(E-CECLL時間で)後にセーブします。トレーサ内でのみ宣言されます。
- 10行目: トレーサに表示させることを100ステップごとに行います。
- 12行目: 501秒間シミュレーションを行います。
- 13行目: セルステートを「sample-after500s.cs」というファイルにセーブします。
- 14行目: シミュレーションを停止します。

tane.ecsに以上の内容が書いてあるので、まずこのファイルを「toy.ecs」としてコピーします。

#### 課題:

では、自分で新しいトレーサを付け足してみましょう。加える物質はSD、E.bc、C.Ebc-SDの3個です。セーブする時間は最初から書いてあるトレーサと共に1000とします。セルステートは「toy-after500s.cs」と変更します。

- それではecellを起動して、今つくったecsファイルを読み込んでみましょう。ReactorPathはecsファイルに記入しているのでコマンドラインで指定する必要はありません。

```
% ./ecell
```

○

#### 課題:

次にecsファイルを「default.ecs」という名前に変更して、もう一度ecellを起動しましょう。ecsファイルを「default.ecs」と名前にして「toy.ecs」の時とどのように変わるでしょうか？

## 4. Saveしたデータを出力、プロットする

ecsファイルで指定した時間にデータがセーブされています。このデータファイルはE-CECLLを実行したディレクトリにDATAというディレクトリが作られ、その下にCYTOPLASM(Pathで指定したもの)というディレクトリができその下に作成されます。これをグラフ表示ツールで表示してみましょう。

- 今回はgnuplotを使用します。

plot

まず、ecdファイルがあるディレクトリに移動します。これはE-CECLLを実行したディレクトリに「Data」というディレクトリが作成されています。その下の「CYTOPLASM」というディレクトリに移動します。

```
% cd Data/CYTOPLASM
```

ecdファイルからgnuplotを使ってプロットするために必要な、準備ファイル作成プログラムをコピーします。ecd2plot1.1.py

### 1. 2Dでプロットしたいときには

```
% ecd2plot1.1.py --xd 2 ecdfile.ecd > gnufilename
% ecd2plot1.1.py ecdfile.ecd > gnufilename
```

オプションは「--xd」(無くても可)をつけ、次にプロットしたいecdファイルを並べ

ます。最後に適当なファイル名をつけて。このファイルに後のgnuplotを使ってプロットするときに必要なデータが記入されます。

ディスプレイに表示しないで、psファイルに直接保存したいときは「—ps psfile.ps」で指定します。

```
% ecd2plot1.1.py —xd 2 —ps psfile.ps ecdfile.ecd > gnufilename
```

2. 3Dでプロットしたいときも2Dのときと同様です。

```
% ecd2plot1.1.py —xd 3 ecdfile.ecd > gnufilename
```

```
% ecd2plot1.1.py —xd 3 —ps psfile.ps ecdfile.ecd > gnufilename
```

上で指定したファイルを用いてプロットします。

```
% gnuplot gnufilename
```

結果は下のような感じになります。

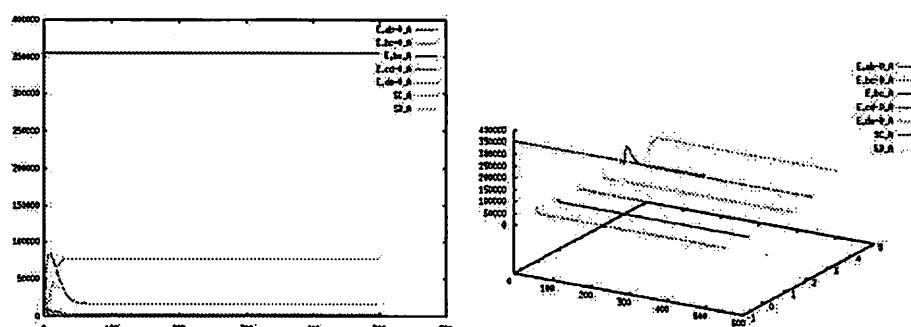


図.4 2Dと3Dでプロットしたもの

以上で第三回は終了です。

今回のチュートリアルに関する感想、質問等は [e-cell-tutorial@eg.e-cell.org](mailto:e-cell-tutorial@eg.e-cell.org) にお願いします。

*moto*  
t97075my@sfc.keio.ac.jp



## 第四回 解糖系を作る ~その1~

### 目標：

### 解糖系のパスウェイマップを作図すること

今回からは、前回までのおもちゃの系ではなく、実在する系をシミュレーションします。題材として *E.coli*(大腸菌)の解糖系を対象にしたいと思います。今回は第一段階としてその系についてE-CELL仕様のパスウェイマップをかきましょう。

1. 解糖系とは何か
2. EcoCycなどを用いてパスウェイマップをかく

#### 1. 解糖系とは何か

解糖とは、グルコース1molがフルクトース1,6-ビスリン酸を経由してピルビン酸2molに変わり、ATPを2mol生産する代謝経路です。この経路は研究されつくした10種の連続酵素反応よりなり、多くの生物のエネルギー供給系であると同時に、グルコースやほかの糖を合成する経路でもあります。

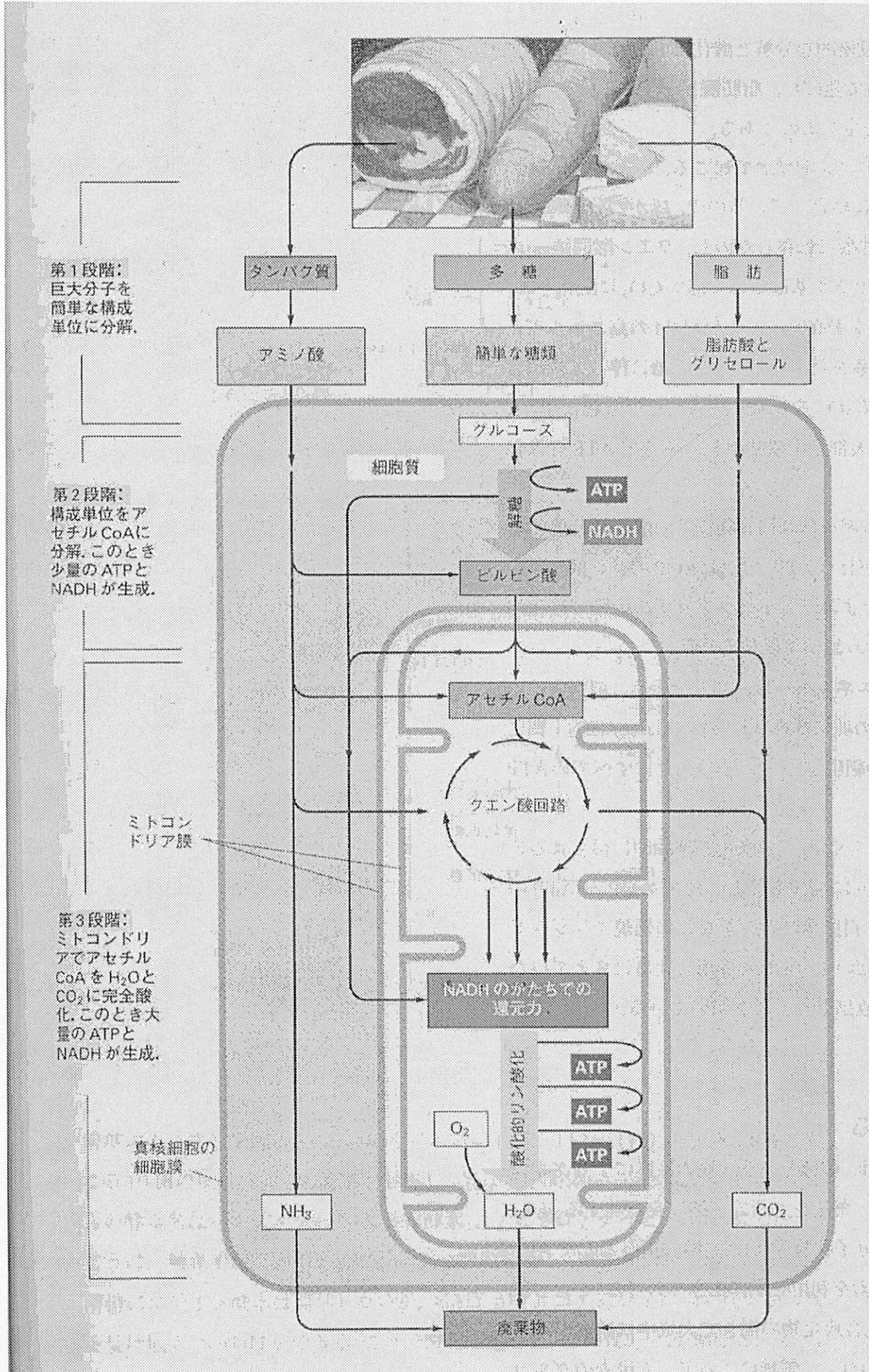
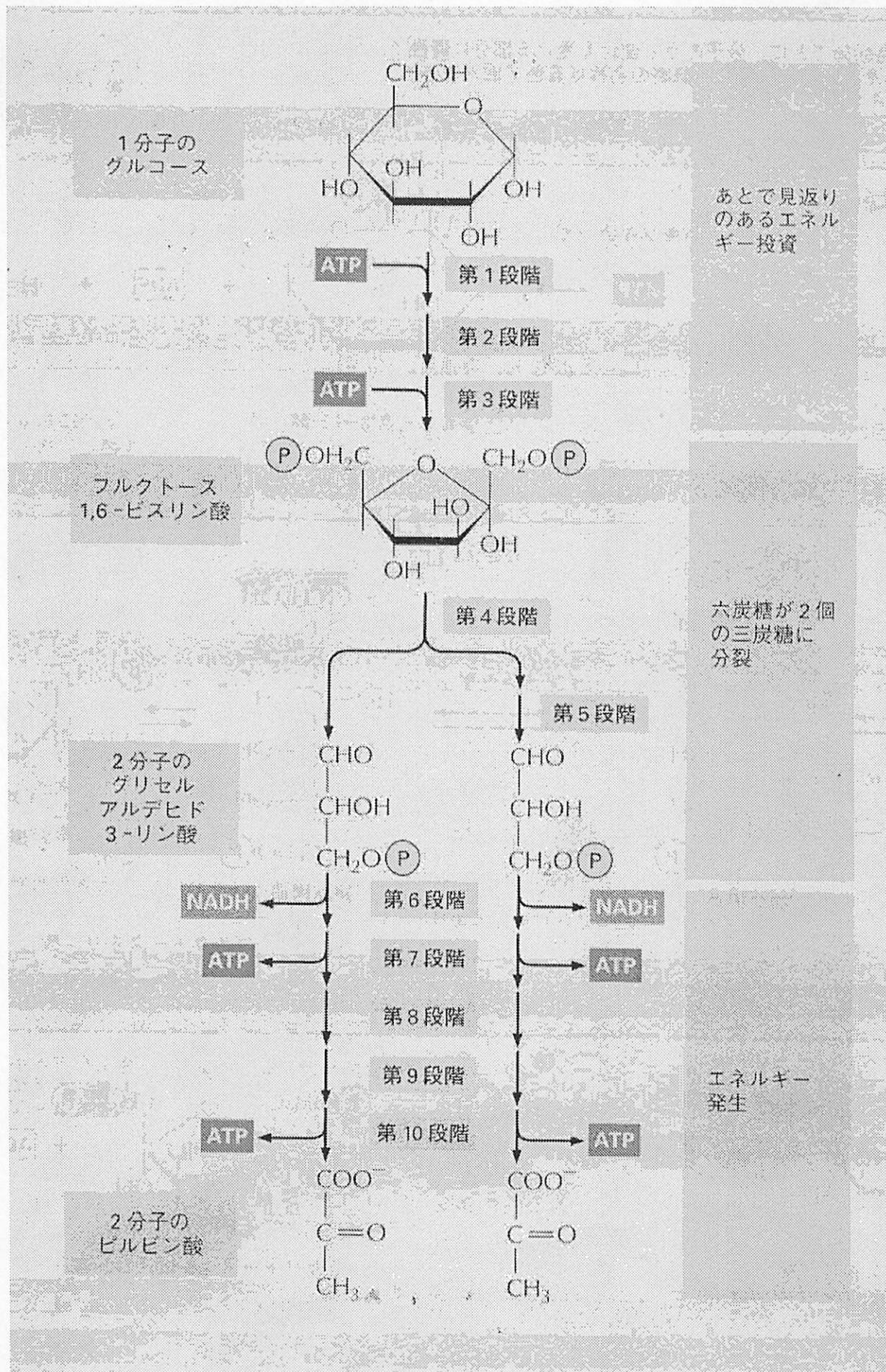


図1:細胞内でのATP生成に関する一連の反応(Essential細胞生物学)



## 図2:解糖経路(Essential細胞生物学)

### 2. EcoCycなどを用いてパスウェイマップをかく

- 解糖系について、実習の第2回で作成したようなパスウェイマップを作成しましょう。
- 解糖系の反応経路図は、EcoCycなどのデータベースから参照することができます。EcoCycとは*E.coli*の全ての分子のカタログを目指して構築されているデータベースで、分子やそれらの相互作用についてのデータが蓄積されています。研究者が遺伝子や代謝経路を柔軟にブラウジングできるグラフィカル・ユーザー・インターフェイスが用意されており、文献データベースやなど、他の生物学データベースへのリンクも充実しています。
- EcoCycで、「Choose from a list of pathways:」から「glycolysis」を選択し、「submit」をクリックします。すると*E.coli* Pathwayが表示されます。ここで「More Detail」を選択できなくなるまでクリックし、もっとも詳しいものにします。その*E.coli*の解糖系パスウェイがこれです。
- さて、E-CELL でシミュレーションを行うためには、シミュレーションの対象とする系を Substance-Reactor モデルで表現する必要があります。Substance-Reactor モデルとは、第3回の講義で説明したように、E-CELL による細胞モデリング手法の基本となる概念です。

EcoCyc に収録されているのは、物質同士の関係をかいた経路図です。これを、Substance-Reactor モデルで表現しやすいように書き換えてみましょう。

Substance-Reactor モデルでは、物質と化学反応をノードとしてプロットします。物質についてはすでにデータベースに記載されているので、これにリアクターにあたる節点を付け加えていけばよいのです。詳しい作図方法については実習の第2回を参考にしてください。

- パスウェイマップができあがったら、リアクターの部分に注目して全体を見直してみましょう。基本的に全て酵素反応ですが、ひとつのリアクターに関わる分子種の数などに違いがあります。

次回は、それぞれの反応についてどのリアクターを使用すればよいかを決めて、解糖系のルールファイルを作成します。

以上で第四回は終了です。

*moto*  
t97075my@sfc.keio.ac.jp



## 第五回 解糖系を作る ~その2~

### 目標：

解糖系のパスウェイマップをもとにルールを書くこと

前回は第一段階として *E.coli* の解糖系について E-CELL 仕様のパスウェイマップをかきました。今回はそのパスウェイマップをもとにスプレッドシートを用いてルールをかきましょう。

#### 1. ルールを書く

##### 1. ルールを書く

- Substanceをリストアップしてみましょう
- IDのつけかたについて  
あとでそのIDがどの物質なのかがわかるようにするということにさえ注意すれば、どのようなものでも構いません
- 反応をリストアップして、それに対するReactorは何を使えばいいか考えましょう
- Star Officeを起動し、Spreadsheetを用いてルールを書きましょう

下のSpreadsheetを参考にしてください

Type	Class	path	ID	Name
System	Cell	/	CELL	The cell
System	Environment	/	ENVIRONMENT	The culture
System	Cytoplasm	/CELL	CYTOPLASM	The cytop
System	Membrane	/CELL	MEMBRANE	The memb
Type	Class	path	ID	Name
Reactor	ConstantParameterReactor	/ENVIRONMENT	VOLUME	Volume in
Reactor	ConstantParameterReactor	/CELL/CYTOPLASM	VOLUME	Volume in
Type	Class	path	ID	Name
Substance		[systempath of super system]	[ID name]	[Substance]
Type	Class	path	ID	Name
Reactor	[Reactor class name]	[systempath of super system]	[ID name]	[Reactor n

続き

Inside	Outside	VolumeIndex
		/ENVIRONMENT:VOLUME
		/CELL/CYTOPLASM:VOLUME
/CELL:CYTOPLASM	/ENVIRONMENT	
Arg_tag	Arg_coeff	init_act
Value	[Volume (liter)]	[usually same as Arg_coeff]
Value	[Volume (liter)]	[usually same as Arg_coeff]
QTY	Memo	
[quantity]		
S_ID	S_path	S_Coeff
[ID name of substrate]	[systempath of supersystem of substrate]	[stoichiometric coefficient of subs]

続き

P_Coeff	C_ID	C_path
[stoichiometric coefficient of product]	[ID name of catalyst]	[systempath of supersystem of catalyst]

以上で第五回は終了です。

*moto*  
t97075my@sfc.keio.ac.jp



# 第六回 解糖系を作る ~その3~

## 目標：

自分の作ったReactorを使ってE-CECLLを動かす

### 1. Reactorを書こう

- Reactorの書き方
- rdファイルとは
- 実際にrdファイルを書く
- dmtool(dmgen)を使えるようになる

### 2. 動かしてみよう

- ルールを完成させる
- ルールファイルをルールインターメディエイトファイルに変換
- ecellを起動する

### 1. Reactorを書こう

- Reactorの書き方
  - ReactorとはSubstanceの量の時間的変化を計算するものです
  - そのソースはC++言語を用いて記述しています
  - E-CECLLでReactorを書くとはrdファイル(filename.rd)を書くことです
  - rdファイルとはReactor Descriptionファイルのこと、Reactorの仕様と実行される処理の内容が書かれています
  - dmtoolを使ってrdファイルはtexファイル(LaTeX形式のReactor Spec Sheet)やsoファイル(E-CELL Systemがロードできる形式)などに変換されます
- rdファイルとは
  - キーワードと値の組からなる行を必要な数を並べて作成するものです
  - rdファイルの構成は大きく
    - 一般情報に関する部分
    - Reactor Spec Sheetに関する部分
    - eactor Source Codeに関する部分
  - の3つに大別できます
  - RDファイルを作成する際には以下の約束事に注意する。
    - キーワードはアルファベットの大文字又は\_で構成され、スペースを含まない。
    - キーワードは @ または % で始まる。@ で始まる行は単純にその内容が読み込まれる。% で始まる行はその中身を,”で区切ることによって、配列

として処理される。

- #で始まる行はコメント文とみなされる。行頭の #をそのまま出力する際は、#とするかスペースを1つ入れる。ただし、行頭以外の #はそのまま出力される。
- 行頭にキーワードのない行は、それより前の行の内容に続くものとして解釈される。(Spec Sheetに変換する際に改行したい場合、改行したい個所に¥¥を入れる。)
- 実際にRDファイルを記述する際は、予めキーワードが一通り書かれて用意されているdefault.rdというファイルに、必要事項を書き込んでいければいいようになっている。
- キーワードは必要でないものに関しては省略できる。

■ rdファイルの例解

1. これはMichaelisUniUniReactorについてのファイルです。このReactorは次の反応速度式に従います。

$$v = \frac{K_{cF}[E][S]}{K_m S + [S]}$$

MichaelisUniUniReactorについてのファイル

@CLASSNAME: MichaelisUniUniReactor

@BASECLASS: FluxReactor

@AUTHOR: E-CECLL Tutorial

@EMAIL: tutorial@e-cell.org

@DATE: 2000 12/12

%VERSION: ecs-v1, 0.1

@BRIEF\_DESCRIPTION: Unireactor enzyme activity of which kinet

@DESCRIPTION: A reactor class for unireactant enzyme activity by the Henri-Michaelis-Menten equation derived from rapid eq  
¥vspace {0.2cm}

This reactor is applicable to the following reaction sequence:

```
¥begin{center}
$E+S ¥rightleftharpoons ^{k_{\{1\}}}_{k_{\{-1\}}} ES
¥rightarrow ^{k_{\{p\}}} E+PS
¥end{center}
¥vspace {0.3cm}
```

@EQUATION: \$\$v=\frac{(K\_{cF}[E][S])}{(K\_m S+[S])}\$\$

%SUBSTANCE: Substrate, 1, 1

%SUBSTANCE: Product, 1, 1

%SUBSTANCE: Catalyst, 1, 1

%SUBSTANCE: Effector, 0, 0

%PARAMETER: KmS, Float, mol/l, Michaelis Constant of Substrate

%PARAMETER: KcF, Float, mol/l, Catalytic Constant (Forward)

```

@REACT_FUNC:
Float S = substrate(0)->concentration();
Float E = catalyst(0)->quantity();

Float velocity = KcF * E * S;
Float Den = KmS + S;

velocity /= Den;
process(velocity);

```

## 2. 一般情報に関するキーワード

- @CLASSNAME: 作成するReactor のクラス名(ファイル名から.rdを削除した形)
- @BASECLASS: そのクラスが継承する元の基底クラス
- @AUTHOR: 作成者名
- @EMAIL: E-Mail address
- @DATE: 作成日
- %VERSION: E-CECLLSYSTEMのバージョン ,このReactorのバージョン
- @BRIEF\_DESCRIPTION: そのReactorの簡単な説明

## 3. Reactor Spec Sheetに関するキーワード

- @DESCRIPTION: その Reactor の詳しい説明
- @EQUATION: Reactor の式を LaTeXのdisplaymath環境で以下のいずれかの方法で記述
  1. \$\$<数式の記述>\$\$
  2. ¥begin[displaymath]<数式の記述>¥end[displaymath]
  3. ¥[<数式の記述>]¥
- %SUBSTANCE: Substanceの定義で、Substrate、Product、Catalyst、Effectorの4種類が用意されています
- @NOTES: Reactorの実装に関する注意点を書きます

## 4. Reactor Source Codeに関するキーワード

- %PARAMETER: パラメータ名、パラメータの型、単位、パラメータに関する記述を、``,"で区切って書きます
- @PRIVATE: オブジェクトの私的要素(できる限りメンバ変数はこれで定義しましょう)
- @PROTECTED: オブジェクトの限定公開要素(派生クラスからは参照できます)
- @PUBLIC: オブジェクトの公開要素
- @INITIALIZE\_FUNC: ここでは主にパラメータの値域のチェックや、シミュレーション中に変化しない値の計算などの初期設定を行います
- @REACT\_FUNC: 每ステップ行う処理を書きます。1.反応速度を算出し、2.その速度にしたがって物質の量を増減させる、といった一連の処理を書きます。Process()メソッドの引数は、1秒あたりの反応分子数で、Float型です。

### ○ 実際にrdファイルを書く

今までルールを書いていたディレクトリに「reactor」というディレクトリを作成し、これからreactor作りはこのreactorというディレクトリの下で行いましょう。

- MichaelisUniUniReactorについてのファイルをコピーしてください。ファイル名は「*CLASSNAME*.rd」です。

- dmtool(dmgen)を使ってrdファイルを変換

1. texファイルに変換しよう

```
% dmtool reactor -d MichaelisUniUniReactor.rd
```

これで「spec.tex」ファイルが生成されていれば成功です

2. soファイルに変換しよう

```
% dmtool reactor -c MichaelisUniUniReactor.rd
```

これで「MichaelisUniUniReactor.so」ファイルが生成されていれば成功です

- それではMichaelisBiBiReactorについても同様にrdファイル、texファイル、soファイルを作成しましょう。このReactorは次の反応速度式に従います。

$$v = \frac{K_c F [E] [S_a][S_b]}{(K_m S_a + [S_a])(K_m S_b + [S_b])}$$

## 2. 動かしてみよう

- ルールを完成させる

頑張って完成させましょう

- ルールファイルをルールインターメディエイトファイルに変換

```
% ss2er < toy.txt > toy.er
```

```
% er2eri < toy.er > toy.eri
```

- ecellを起動する

以上で第六回は終了です。

*moto*

t97075my@sfc.keio.ac.jp



## 第七回 パラメータチューニング

### 目標:

パラメータチューニングができるようになる

シミュレーション結果は、初期値や反応速度定数など、モデルのパラメータの値に左右されます。本来その値は実験で求められた正確な値であるべきです。しかし実際には、シミュレーションに必要なパラメータについての信頼すべきデータが完全にそろうなんていう幸運に恵まれることはあまりありません。講義第6回では、こうした未知のパラメータの値を探索する方法を学びました。

実習では、その手法のひとつ「Genetic Algorithm」を E-CELL に適用するプログラムを用いて、以前に作った「toy pathway」モデルが望ましい動きをするようなパラメータの値を探します。

必要なものは以下のとおりです。

1. 探索したいパラメータの値だけをマスクした E-CELL のルールファイル
2. そのパラメータの値を正しくセットすることで得られるはずのシミュレーション結果
3. パラメータチューニングを行なうプログラム
4. プログラムに与える条件を記述したファイル

2と3は既に用意してあります。1と4について、雛型に変更を加えて完成させるのが今日の実習の具体的な課題です。

- 課題1 道具をそろえましょう

実習に必要なファイルを圧縮した “estimation.tgz” を自分の作業領域にもってきて、展開してください。「estimation」というディレクトリができます。今回の作業はここで行ないます。

```
tar xzf estimation.tgz
cd estimation
ls
```

だいたいこんなものが入っています。

- toy.er.in: 探索したいパラメータの値だけをマスクした er ファイル(用意するもの1)  
パラメータチューニングのプログラムはこれを読みこんで、様々なパラメータ値をあてはめた er ファイルを生成して何度もシミュレーションを行ないます。  
実習の第2、3回で作成した toy.er ファイルを、toy.er.in ファイルとしてコピーして使って構いません。
- Data/CYTOPLASM/ほにや.ecd.: そのパラメータの値を正しくセットしたことで得られるはずのシミュレーション結果(用意するもの2)  
.ecd(E-CELL data)の形式でなくてはいけません。

- GA: パラメータチューニングを行なうプログラム(用意するもの3)  
実体は、GAsrc/ 以下にあります。
- default.pes: プログラムに与える条件を記述したファイル(用意するもの4)
- 課題2 探索するパラメータを指定する

toy.er.in を書きかえましょう。

用意してある toy.er.in は、「toy pathway」のルールファイルと同じです。.er.in ファイルは、だいたい普通の er ファイルと同じですが、チューニングしたいパラメータについて、探索範囲を指定することができます。探索するパラメータを複数指定することもできますが、数が増えれば計算時間も増します。今回はひとつにしましょう。反応速度定数のどれかにするのがよいでしょう。

ターゲットにするパラメータについて、その値を書いてある部分を、

[数値1,数値2]

という形に書きかえてください。

こうしておくと、このファイルを読んだパラメータチューニングのプログラムが、数値1と数値2の範囲でパラメータ値を探してくれます。

例: 「KcF」というパラメータの値を 0.1 から 5.3 の範囲でさがすなら、、

KcF [0.1, 5.3];

- 課題3 .pes を書きかえる

パラメータチューニングのプログラムに渡すデータのほとんどは、.pes ファイル(parameter estimation script)ファイルに記入します。

pes ファイルのサンプル

```

method : GA
ecsfile : sample.ecs.in
data_dir : ./Data/CYTOPLASM/
ecell : ecell
er2eri : er2eri
result_dir : ./result
result_file : result.ecd
population : 10
max_locus : 30
elite_strategy : off
scaling_type : inverse
max_generation : 10
reproduction_type : roulette_strategy
crossover_type : one_point_2
saturation_g : 1000
generation_gap : 0.9
crossover_p : 0.9
mutation_p : 0.01

```

min_variety	: 0.01
epsilon	: 0.01
theta	: 0.000001
aline	: 0

「ecsfile」で指定した .ecs.in ファイルには、望んでいるシミュレーションデータ（「data\_dir」で指定している Data/CYTOPLASM/ の下にある）に対応するデータを毎回セーブするためのスクリプトや、探索一回分の時間、.er.in から自動生成される .eri ファイルの指定が記入されています。この例は用意された sample.ecs.in を読むように指示しています。今回の実習ではこれを変更する必要はありません。

「population」から先は、GA プログラムに与えるパラメータ値探索の条件の指定です。それぞれの条件の意味について詳しくは講義を参照してください。どの条件をどのように設定するかでパラメータチューニングの成功率や正確さが大きくかわりますから、この部分の吟味には重点を置く価値があります。

- population: 個体数
- max\_locus: 遺伝子長(変えてあまり変化しない)
- elite\_strategy: エリート戦略
  - elite
  - semi\_elite
  - off
- scaling\_type: 個体の適応度
  - linear
  - root
  - log
  - ln
  - inverse
- max\_generation: 最大繰り返し数
- reproduction\_type: 再生規則
  - roulette\_strategy
- crossover\_type: 交叉
  - one-point: 1点交叉—交叉ポイントを選び、その後の配列を交換
  - one-point\_2: 1点交叉—交叉ポイントを選び、その前の配列を交換
  - two-point: 2点交叉
  - uniform: 一様交叉
- saturation\_g: エリートが何世代変化しないと終了するか
- generation\_gap: 世代間でどのくらい変化するか
- crossover\_p: 交叉率
- mutation\_p: 突然変異率
- min\_variety: 終了条件

- epsilon: 終了条件( $\epsilon$ )
  - theta: 終了条件( $\theta$ )
    - alien: 新しい個体をどのくらい作るか  
では、default.pes のいくつかの条件を変更してセーブしてください。
  - 課題4 プログラムを走らせよう
- GA によるパラメータチューニングを行なうプログラム、“GA”を実行します。引数として、.er.in ファイル名を与えます。

```
./GA toy.er.in
```

パラメータ値の探索が終わるとプログラムは終了し、どの終了条件を満たして探索を終えたのかが表示されます。終了条件は以下のどれかです。

1. ある個体の評価関数値が0になった。

$$F_{min}(g) = 0$$

2. ある世代gのエリート個体の評価関数値  $F_{min}(g)$  が第1世代のエリート個体の評価関数値に比べて十分に小さくなった。つまり、

$$\frac{F_{min}(g)}{F_{min}(0)} < \theta$$

を満たした。(θは0.0001程度の値)

3. 数世代  $g_s$  に渡って、評価関数値がほとんど変化しなくなった(飽和した)。つまり、

$$\frac{F_{min}(g - g_s) - F_{min}(g)}{F_{min}(g)} < \epsilon$$

を満たした。(εは0.01程度の値)

4. 個体間の多様性がなくなった。つまり、

$$\frac{F_{max}(g) - F_{min}(g)}{F_{max}(g)} < v$$

を満たした。(vは0.01程度の値)

5. 世代交替の回数があらかじめ決められた最大繰り返し数  $g_{max}$  に達した。

$$g = g_{max}$$

適切な条件を満たして終了している場合(2、4、3 の順にいい結果)は成功です。そうでない場合は、もう一度探索の条件を見直してみるほうがいいかもしれません。例えば、1ならバグの可能性を疑いましょう。ふつうありえないことだからです。5は、いい結果が得られないうちに最大繰り返し数("max\_generation"で指定)に達してしまったという意味ですから、"max\_generation"をもっと高い値にするなどの修正が考えられます。

- おまけ課題 答えの時系列データとチューニング後のデータを比べて確認する

"GA"が終了すると、チューニングして得られたパラメータ値のセットされた .eri ファイルが生成されています。今回は toy.er.in をプログラムにかけたので、toy.eri ができます。余力のある人は、E-CELL でこのルールを読みこんでシミュレーションを行ない、望んでいたシミュレーション結果と近いかどうか確認してみましょう。./Data/CYTOPLASM/ 以下に用意したデータは、シミュレーション開始後 50 秒までのものです。やり方はいろいろあります。ではがんばってな。

ヒントになる実習：

第3回のシミュレーションデータのセーブとグラフ化、.ecs の書き方など。

注意：

ふつうにトレーサーからデータを保存すると、E-CELL を実行したディレクトリの下の Data/CYTOPLASM/ 以下にデータが保存されます。「estimation」ディレクトリで確認作業をする場合は、気をつけてください。

以上で第七回は終了です。

*moto*

t97075my@sfc.keio.ac.jp



## INDEX

**1 はじめに****2 基本的な操作の流れ**

- 2.1 E-CELL Systemの起動
- 2.2 ルールファイルの読み込み
- 2.3 スクリプトファイルの読み込み
- 2.4 シミュレーションの実行
- 2.5 ステップ幅の変更
- 2.6 シミュレーションの結果の表示
  - 2.6.1 Tracerによる表示
  - 2.6.2 SubstanceWindowによる表示
  - 2.6.3 ReactorWindowによる表示
- 2.7 シミュレーションの結果の保存
- 2.8 E-CELL Systemの終了

**3 各インターフェイス**

- 3.1 Control Panel
- 3.2 ファイル選択ウィンドウ
- 3.3 Tracer
- 3.4 EntrySelector
- 3.5 SubstanceWindow
- 3.6 ReactorWindow

**4 E-CELLスクリプトファイル**

- 4.1 スクリプトファイルとは
- 4.2 スクリプトファイルの文法について
- 4.3 スクリプトファイルのサンプル
- 4.4 サンプルファイルの解説
- 4.5 スクリプトファイルの命令の一覧

## 1 はじめに

この章ではE-CELL Systemの基本操作について説明する。まず、``2節 基本的な操作の流れ''では、簡単にE-CELL Systemの起動から終了について述べる。細かい操作に関しては次の``3節 各インターフェイス''の説明で述べられている。また、E-CELL Systemの操作性を向上させるスクリプトファイルに関しては、別に``4節 E-CELLスクリプトファイル''で詳しく解説している。

また、E-CELL Systemはシミュレーションを行うために、データファイルであるルールファイルを必要とし、このルールファイルの詳細については2章で解説する。また、新たなモデリングを行うためには、新たなReactorの構築を要することがあり、このReactorについては3章で詳細に解説する。これらを全て通読すれば、基礎的なE-CELL Systemでのモデリングが可能になるであろう。

## 2 基本的な操作の流れ

## 2.1 E-CELL Systemの起動

E-CELL Systemを起動するには、シェル上でE-CELL Systemの実行可能ファイル (eCell)があるディレクトリーに、パスが通っている状態で、eCellと入力する。

E-CELL Systemが起動するとまず、E-CELL Systemからのエラー等のメッセージを表示するための MessageWindowが表示され、次にE-CELL Control Panelが表示される。

E-CELLを起動した時のカレントディレクトリーに、default.ecsという名前のスクリプトファイルがあると、そのファイルは自動的に読み込まれる。ただし、このスクリプトファイル内にLoadRule命令がなく、Interfaceの呼び出しありRun命令があると、エラーにより実行不能となるので注意すること。また、起動時にオプションをつけることにより、いくつかの設定とファイルの読み込みを行える。そのオプションを以下に示す。

- a *classname* 省略時アクチュエータークラスを指定する(2章参照)。  
このオプションが指定され、しかもシミュレーション経過時間が0の場合に、セルステートファイルを読み込むと、そのセルステートファイルに記録されているシミュレーション経過時間が、現在のシミュレーション経過時間に設定される。他の場合には、セルステートファイルを読み込んでも、シミュレーション経過時間は変化しない。
- c デバッグモードで起動する。
- d スクリプトファイルを読み込む。
- f *filename* オプション一覧を表示する。
- h Reactorの実行ファイルのあるディレクトリのパスを指定する。
- r *directory\_name* 一時ファイルの書き込み先を指定する。
- t *directory\_name* 整数値 初期化時にPostern Reactorを呼出す回数を指定する。詳細については、3章のPostern Reactor参照。

## 2.2 ルールファイルの読み込み

ルール(erl)ファイルは、E-CELL Systemがシミュレーションを行うのに必要な情報が記述されたデータファイルであり、シミュレーションを行う前に必ず読み込む必要がある。ルール(erl)ファイルを読み込むには、ControlPanelのFileメニューよりLoad Ruleを選択し、新たに表示されるファイル選択ウィンドウから目的のルール(erl)ファイルを選択する。

なお、ルールファイルを2回以上読み込んではいけない。(ルールファイルを2回読み込もうするとE-CELLが異常終了するという問題については、次バージョンでの改善を予定している。)

## 2.3 スクリプトファイルの読み込み

E-Cell起動時のカレントディレクトリーにdefault.ecs という名前のファイルがあると、起動時に自動的に読み込まれる。このファイルを読み込む途中でエラーが起きると、E-Cellが異常終了する可能性があるので、注意を要する。

スクリプト(ecr)ファイルはシミュレーションや、各インターフェイスを制御するためのファイルである。スクリプト(ecr)ファイルは必ずしも読み込む必要はない。スクリプト(ecr)ファイルを読み込むには、ControlPanelのFileメニューよりLoad Scriptを選択し、新たに表示されるファイル選択ウィンドウから、目的のスクリプトファイルを選択する。

起動時オプション-c参照。

## 2.4 シミュレーションの実行

ルールファイルを読み込んだ後、ControlPanelのStartボタンをクリックすると、シミュレーションが開始され、シミュレーション系内での時間の経過がタイムカウンターに表示される。シミュレーションを停止するには、Stopボタンをクリックする。また、その隣にあるStepボタンをクリックすると、1ステップ時間幅分だけのシミュレーションが行われる。

## 2.5 ステップ幅の変更

ステップとは、反応の進む速度を計算し、Substanceの量を更新する1サイクルのことである。デフォルトでは1/1000秒を1ステップの積分時間幅としている。ステップ時間幅変更ボタンをクリックし、新たに表示されるテキストボックスに数値を入力して、この時間幅を変更できる。また、ecsファイルによっても指定できる。

ただし、時間幅を大きくすると、シミュレーションにかかる時間は短縮されるが、誤差が大きくなり、逆に、時間幅を小さくすればシミュレーションにかかる時間は長くなるが、誤差を小さくすることができる。

## 2.6 シミュレーションの結果の表示

初期設定では、100回の積分計算につき1回、ユーザーインターフェースが更新される。ecsファイルで、この間隔を変更できる。

### 2.6.1 Tracerによる表示

Tracerはシミュレーションの結果を、Substanceの分子数もしくは濃度の変化や、Reactorの活性の変化と、時間との2次元グラフとして表示する。新たなTracerWindowを表示させるには、ControlPanelのNewInterfaceメニューよりTracerを選択する。Tracerに新たなSubstanceやReactorを表示させるには、下にあるAddボタンをクリックする。すると、EntrySelectorが新たに表示されるので、そこから、目的のSubstanceかReactorを選択する。

なお、Substanceが表す物は、厳密な化学用語における「分子」とは限らず、イオン、ラジカル等の任意の粒子でありうるが、本マニュアルでは便宜的に「分子」という用語を使う。

選択されたSubstanceやReactorは、Traces表示部に色つきの正方形のトグルボタンとともに表示される。その状態で、シミュレーションを開始すると、そのSubstanceもしくはReactorの状態が、そのトグルボタンの色と同じ色のグラフで描画される。

### 2.6.2 SubstanceWindowによる表示

物質量を表示したり変更するにはSubstanceWindowを用いる。SubstanceWindowを表示させるためには、ControlPanelのNewInterfaceメニューよりSubstanceWindowを選択する。すると、EntrySelectorが新たに表示されるので、そこから目的のSubstanceを選択し、OKボタンをクリックするとSubstanceWindowが表示される。

SubstanceWindowには上から、ID、物質名が表示されており、その下の左側に分子数が、またその右に濃度が表示されている。シミュレーションが停止した状態ならばこの二つの値は、直接変更でき、変更したい値の表示部をクリックして値を入力した後、リターンキーを押すことで変更できる。また、分子数は右にある上下ボタンでも量を変化させることができる。

### 2.6.3 ReactorWindowによる表示

反応の速度や、それに関わる基質、生成物等を見るには、ReactorWindowを用いる。ReactorWindowを表示させるためには、ControlPanelのNewInterfaceメニューより ReactorWindowを選択する。するとEntrySelectorが新たに表示されるので、そこから目的のReactorを選択し、OKボタンをクリックするとReactorWindowが表示される。

ReactorWindowには上から、Reactorクラス、エントリーネーム、その反応の名前が表示され、その下に秒あたりの活性度が表示される。

この表示は簡略な表示モードであり、反応にかかる基質や、生成物等を見たいときには右上にある、ShowListボタンをクリックする。すると、下に新たな表示領域が加わり、それぞれ左から反応の基質のリスト、反応の生成物のリスト、触媒のリスト、反応に影響を与える物質のリストとなっている。

## 2.7 シミュレーションの結果の保存

Tracerに表示されたシミュレーションの結果を保存したいときには、TracerWindowの左下にあるSaveボタンをクリックする。すると、E-CELL Systemを実行されたディレクトリの下に Dataというディレクトリが作られ、さらにSystemの階層構造に応じてサブディレクトリが作られ、SubstanceまたはReactorごとに、ecd形式のデータファイルができる。ecd形式のデータファイルは、一般的には、次の表のようなヘッダー、ヘッダーの終りを示す5個以上の「-」、データ本体、終了を示す「///」から成る。

Header of an ecd file

key word	value
DATA	name of data
SIZE	number of columns and rows
LABEL	legend of each column
NOTE	optional comment

E-CELLが出力したecdファイルの例を示す。

DATA: /CELL/CYTOPLASM:A:Quantity

SIZE: 2 12

LABEL: time quantity

NOTE:

---

0	20000
0. 101	20000
0. 201	20000
0. 301	20000
0. 401	20000
0. 501	20000
0. 601	20000
0. 701	20000
0. 801	20000
0. 901	20000
1. 001	20000
1. 101	20000

///

また、シミュレーションを一時中断したいなど、シミュレーションを停止してモデル全体の物質の状態を保存したい時には、ControlPanelのFileメニューよりSaveCellStateを選択し、適当な名前に.csという拡張子をつけて保存する。ここで保存されたモデルの状態は、同じFileメニューの

LoadCellStateを用いて、読み込むことがき、その時点からのシミュレーションを再開できる。

また、X-Windowの異常、シグナルの受信、演算例外などでプログラムの実行を続けられなくなると、E-CELLは、カレントディレクトリに、*rescure-process\_id.cs*という名前のセルステートファイルを保存してから終了しようと試みる。

## 2.8 E-CELL Systemの終了

ControlPanelのFileメニューよりQuitを選択すると、確認のダイアログが表示され、OKボタンをクリックするとE-CELL Systemが終了する。

# 3 各インターフェイス

## 3.1 Control Panel

Control Panelは、ファイルの読み書き、他のインターフェイスの操作、シミュレーションの実行、停止といった操作を行う E-CELL Systemの中心となるインターフェイスである。

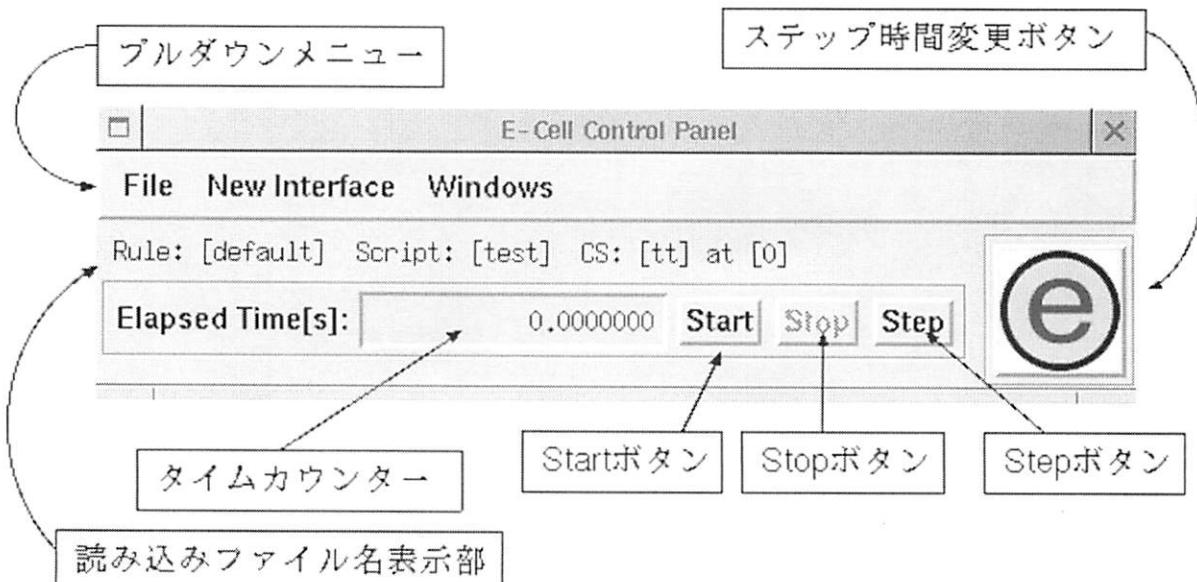


図.1 Control Panel

### プルダウンメニュー一欄

File	Load Rule	ルールファイルを読み込む。
	Load Script	スクリプトファイルを読み込む。
	Load Cell State	Cell Stateを読み込む。
	Save Cell State	Cell Stateを保存する。
	Quit	E-CELL Systemを終了する。
NewInterface	Tracer	TracerWindowを表示する。
	Substance Window	SubstanceWindowを表示する。
	Reactor Window	ReactorWindowを表示する。

Windows Show Message Window MessageWindowを表示する。  
 Hide Message Window MessageWindowを隠す。

Startボタン  
 シミュレーションを実行する。

Stopボタン  
 シミュレーションを停止する。

Stepボタン  
 シミュレーションを1ステップ実行する。

ステップ時間変更ボタン  
 シミュレーションの積分時間間隔を変更するためのダイアログを表示する。

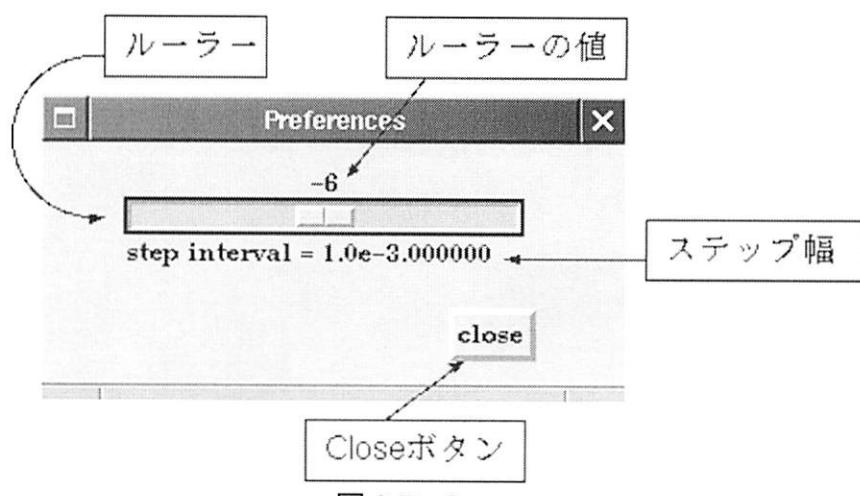


図.2 Preference

テキストボックスに数値を入力して、積分時間間隔を指定する。単位は秒である。省略値は0.001である。

タイムカウンター  
 シミュレーションシステム内の経過時間を秒単位で表示する。

### 3.2 ファイル選択ウィンドウ

ファイル選択ウィンドウはファイルの読み書きを行う際に、ファイルを選択するために表示される。通常は、左にあるディレクトリ表示部のディレクトリ名を、目的のディレクトリに達するまでクリックした後、右のファイル表示部より、目的のファイルをクリックし、下のOKボタンをクリックして、ファイルを選択できる。また、新規のファイルに書き込みを行いたいときは、selection表示部をクリックし、直接ファイル名を入力したのち、OKボタンをクリックする。

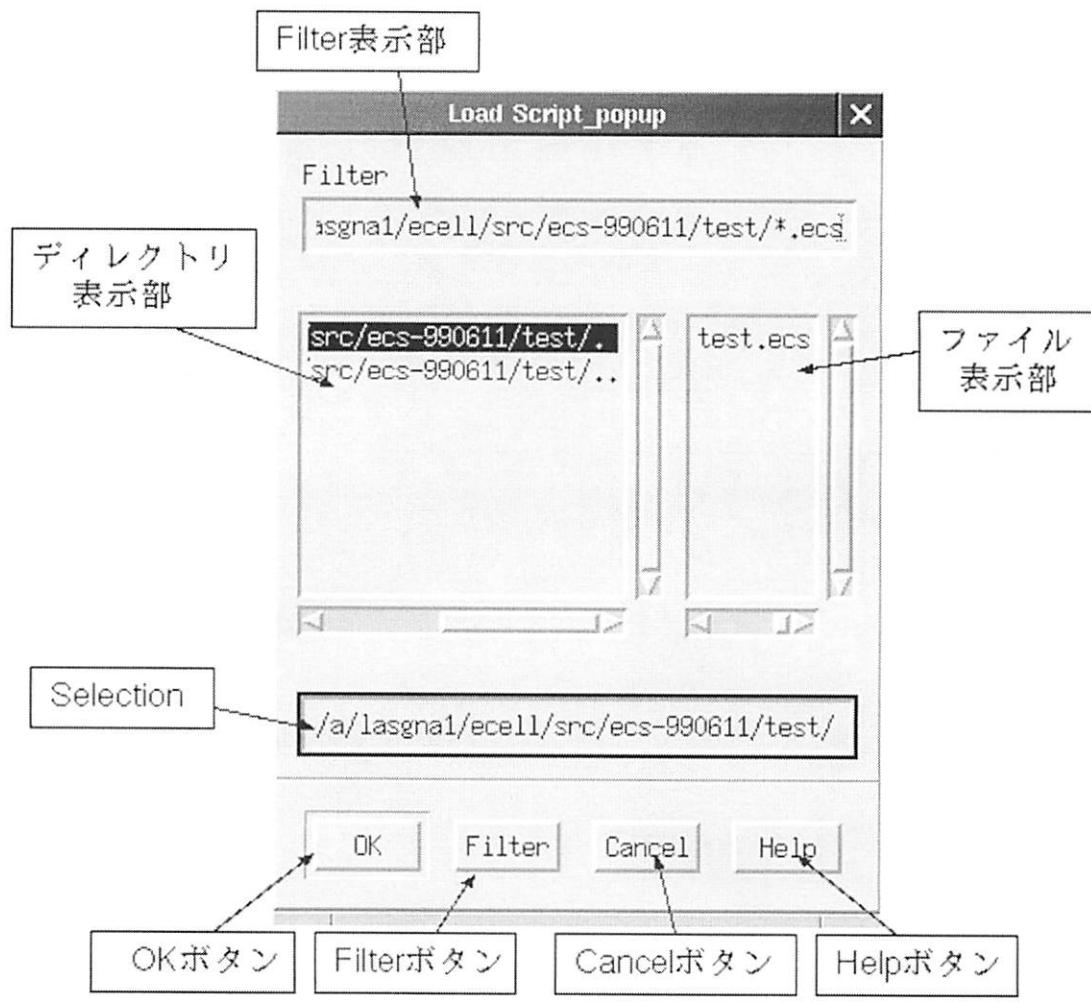


図.3 FileSelector

#### Filter

Files表示部に表示されるファイルの表示をフィルターにより制限するため、正規表現を表示する。初期値では、\*.ecs、\*.csのように、現在選択されるべきファイルの拡張子を持つファイルのみが表示されるようになっている。フィルターを変更するには、表示部をクリックし、入力後、リターンキーを押すか、下にあるFilterボタンをクリックする。親ディレクトリーに移るためには、この場所のファイル名を消して、「..」を書き込んでからEnterボタンを押す。

#### Directories

現在のディレクトリーから移動できるディレクトリーの一覧である。表示されているディレクトリ名をクリックすると、上にあるFilter表示部のディレクトリー表示がクリックされたディレクトリーのものとなり、Filterボタンをクリックすることで、そのディレクトリーに移動し、Files表示部のファイル表示がそのディレクトリーのものとなるまた、ディレクトリ名をダブルクリックしても、ディレクトリーを移動できる。

#### Files

現在選択可能なファイルの一覧である。表示されているファイル名をクリックすることにより、下にあるSelection表示部のファイル名がクリックされたファイルのものとなり、OKボタンをクリックすることで、そのファイルが選択される。また、ファイル名をダブルクリックしても、ファイルを選択できる。

#### Selection

現在選択されているファイル名をフルパスで表示する。表示部をクリックすることで、ファイル名を直接入力できる。入力後、リターンキーを押すか、OKボタンをクリックすると、そのファイルが選択される。

#### OKボタン

クリックすると、現在、Selection表示部に表示されているファイルが選択される。

#### Filterボタン

クリックすると、現在、Filter表示部に表示されているように、ファイル表示部にフィルターがかけられる。

Cancelボタン

クリックするとファイルの選択を終了し、ウィンドウが閉じる。

Helpボタン

現在のバージョンでは使用できない。

### 3.3 Tracer

TracerはSubstanceの物質量のまたは Reactorの活性の変化を、時間との2次元のグラフで表示する。1個のトレーサーウィンドウにつき、最大8系列のデータを表示できる。

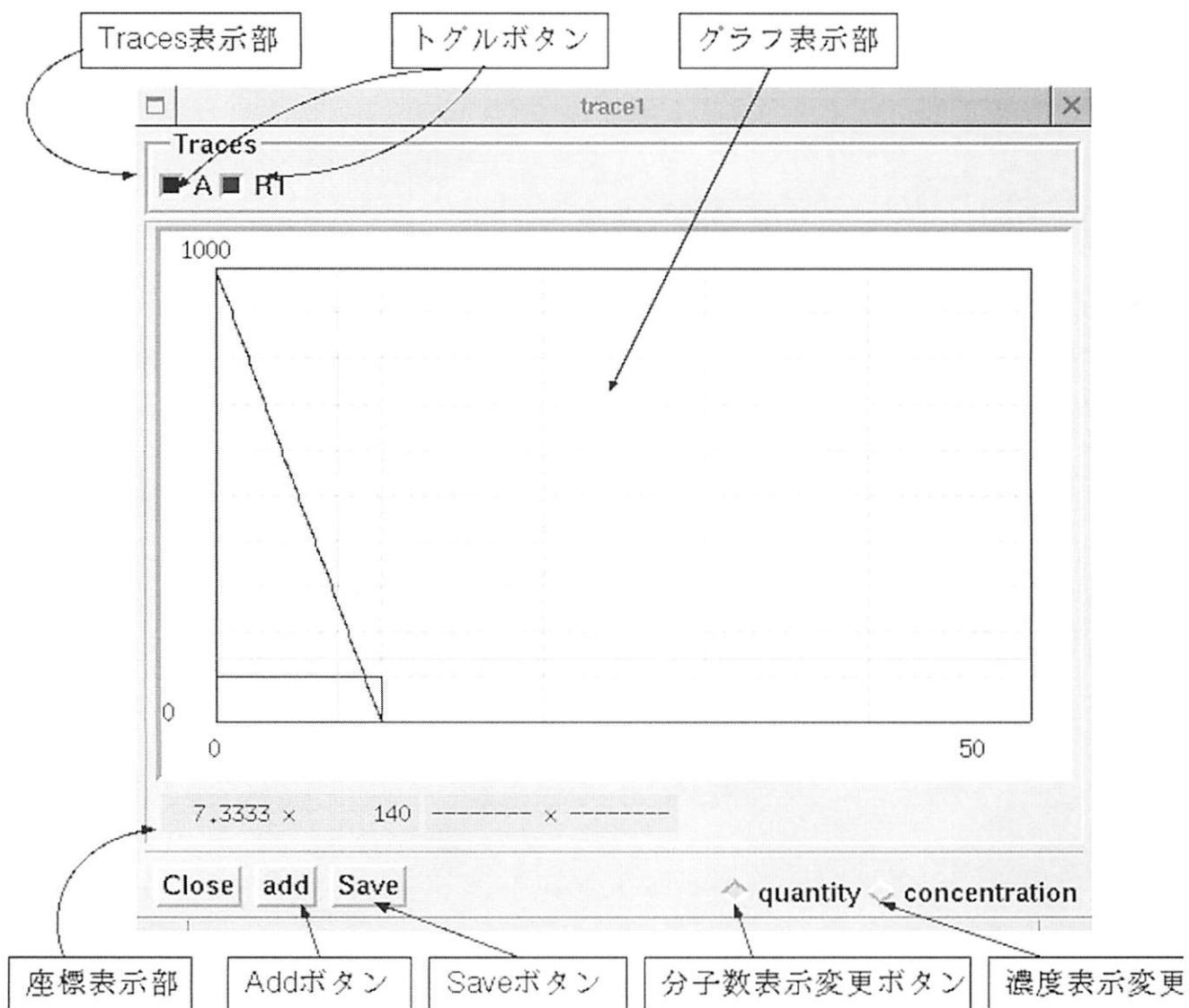


図.4 Tracer

#### Traces

Tracerに表示されるSubstanceとReactorの一覧が表示される。名前の左のラジオボタンの色は下のグラフの色と対応しており、ボタンの左クリックで、グラフの表示、非表示を切り替え、右クリックにより、それがSubstanceならば、SubstanceWindowが、ReactorであればReactorWindowが表示される。1個のトレーサーには、合計最大8個のSubstanceまたはReactorを表示できる。

#### グラフ表示部

シミュレーションの結果がグラフで表示される。縦軸が分子数(もしくは濃度)、横軸が時間を表す。グラフ上で左クリックをするとクリックした点を交点とする青い十字が表示される。この青い十字は、複数のTracerWindowが表示されている際は、全て同一の座標に表示される。この青い十字を消したい時はグラフ表示部外をクリックすればよい。また、中ボタンのドラッグにより矩形を描くことで、その範囲を拡大して表示できる。この拡大した状態をキャンセルするには右クリックをすればよい。また、Shiftボタンを押しながら、グラフ表示部にマウスカーソルを入れると、通常のマウスカーソルの替わりに黒い十字カーソルが表示される。

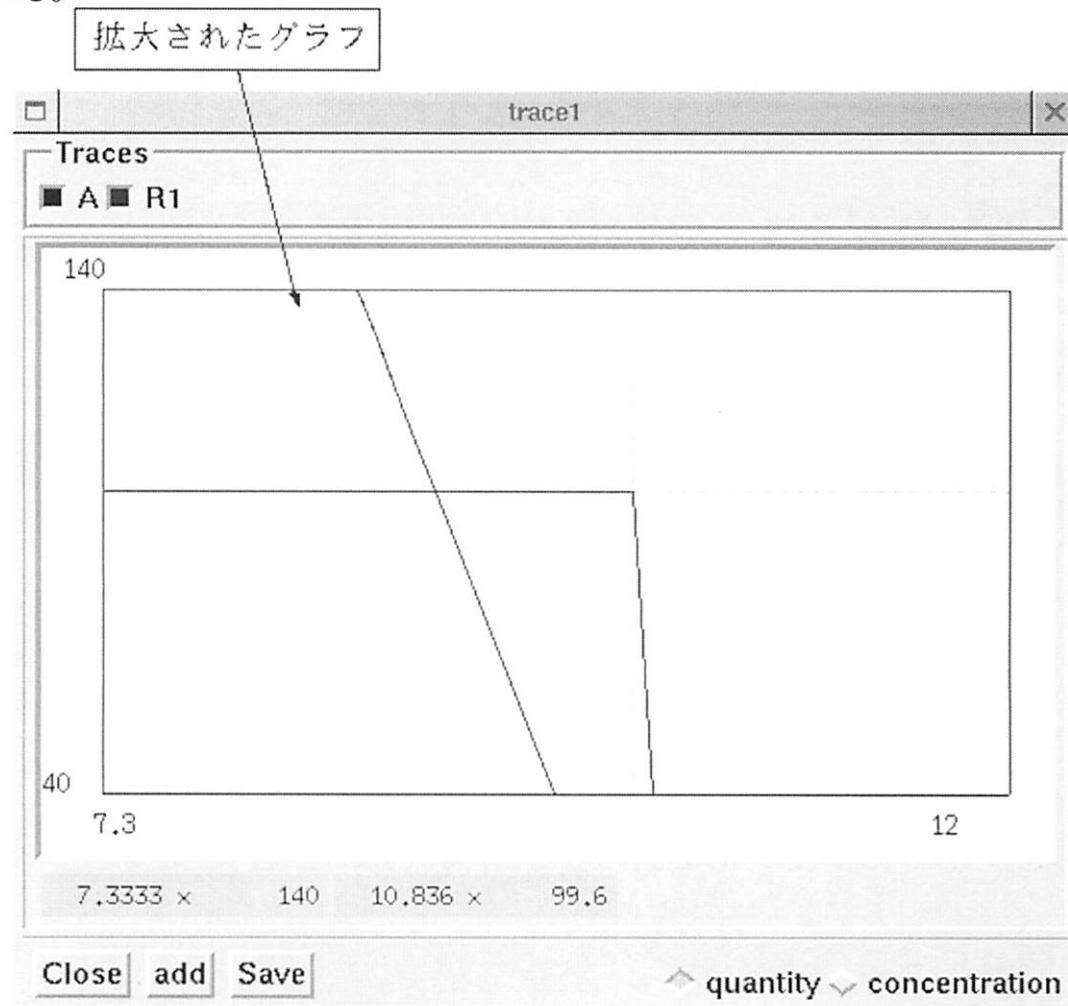


図.5 Tracer(拡大表示)

#### 座標表示部

左には上のグラフ上で左クリックをしたときに表示される青い十字の座標が、右はマウスカーソルの座標が表示される。

#### Closeボタン

TracerWindowを閉じる。

#### Addボタン

新たなEntrySelectorを表示し、そこで選択された新たな SubstanceもしくはReactorを Tracerに表示する。

#### Saveボタン

Tracerの対象となっている時系列データをファイルに保存する。

## 3.4 EntrySelectorの説明

EntrySelectorはSubstanceやReactorを選択する際に使われる。左のSystem選択ダイアログのフォルダアイコンをクリックし、目的のSystemが表示されたら、右のSubstance,Reactor選択ダイアログより、目的のものを選択し、下のOKボタンを押す。また、Tracerより呼ばれた際には、右上の切

り替えボタンにより、選択するものがSubstanceかReactorかを先に選択する。

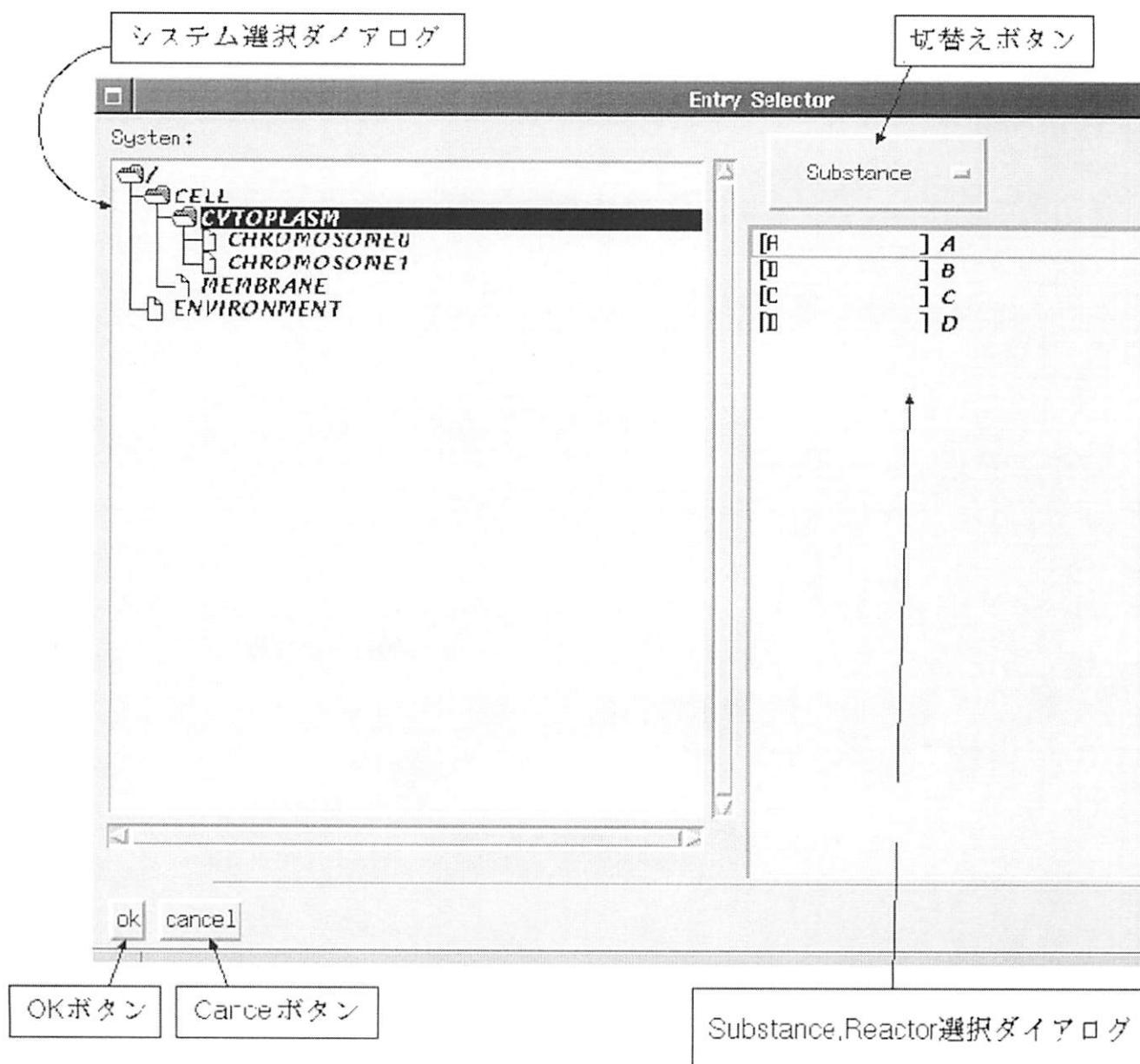


図.6 EntrySelector

#### 切り替えボタン

Tracerより呼び出された際に、SubstanceとReactorを切りかえるときに使う。

#### System選択ダイアログ

Systemが階層構造で表示される。下の階層を表示させたいときにはフォルダアイコンをクリックする。

#### Substance、Reactor選択ダイアログ

SubstanceもしくはReactorが表示される。ドラッグすると複数の選択が可能である。また、Ctrlキーを押しながらクリックすると離れた場所にあるものも複数選択できる。

#### OKボタン

選択を決定する。

#### Cancelボタン

今までの選択を取り消して、ウインドウを閉じる。

### 3.5 SubstanceWindow

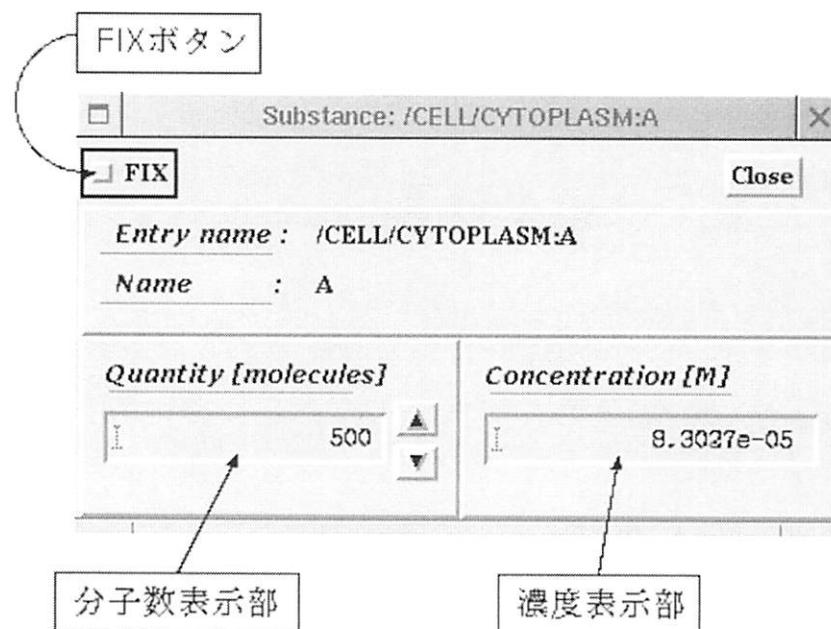


図.7 Substance Window

#### FIXボタン

ボタンが赤色になっている場合には、シミュレーションによって 分子数が変化しない。トグルボタンである。

#### 分子数表示部

そのSubstanceの分子数を表示する。シミュレーションが停止しているときに、クリックすることで値を変えられる。

#### 濃度表示部

そのSubstanceの濃度を表示する。シミュレーションが停止しているときに、クリックすることで値を変えられる。

### 3.6 ReactorWindow

#### 通常表示モード

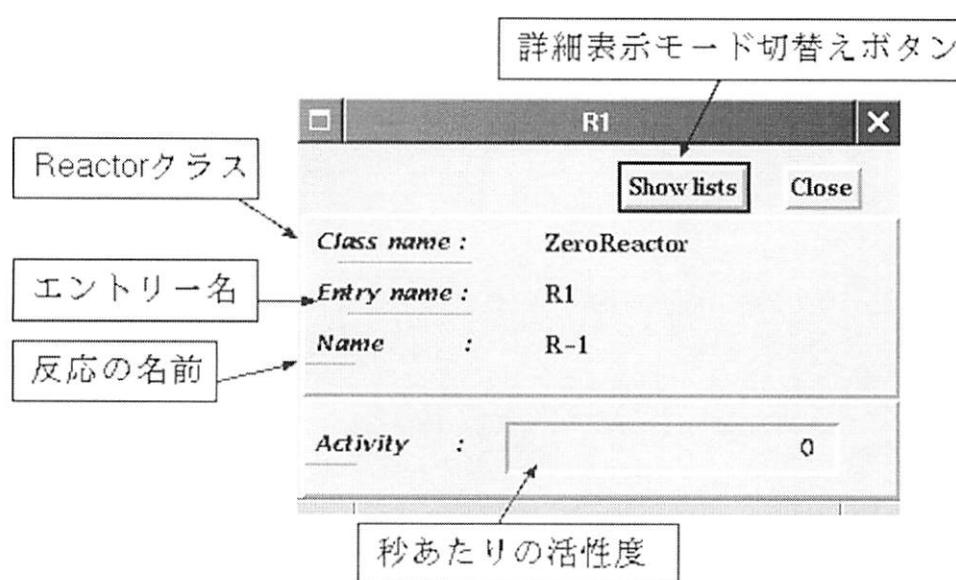


図.8 Reactor Window(通常表示)

詳細表示モード切り替えボタン  
表示を詳細表示に切り替える。

### 詳細表示モード

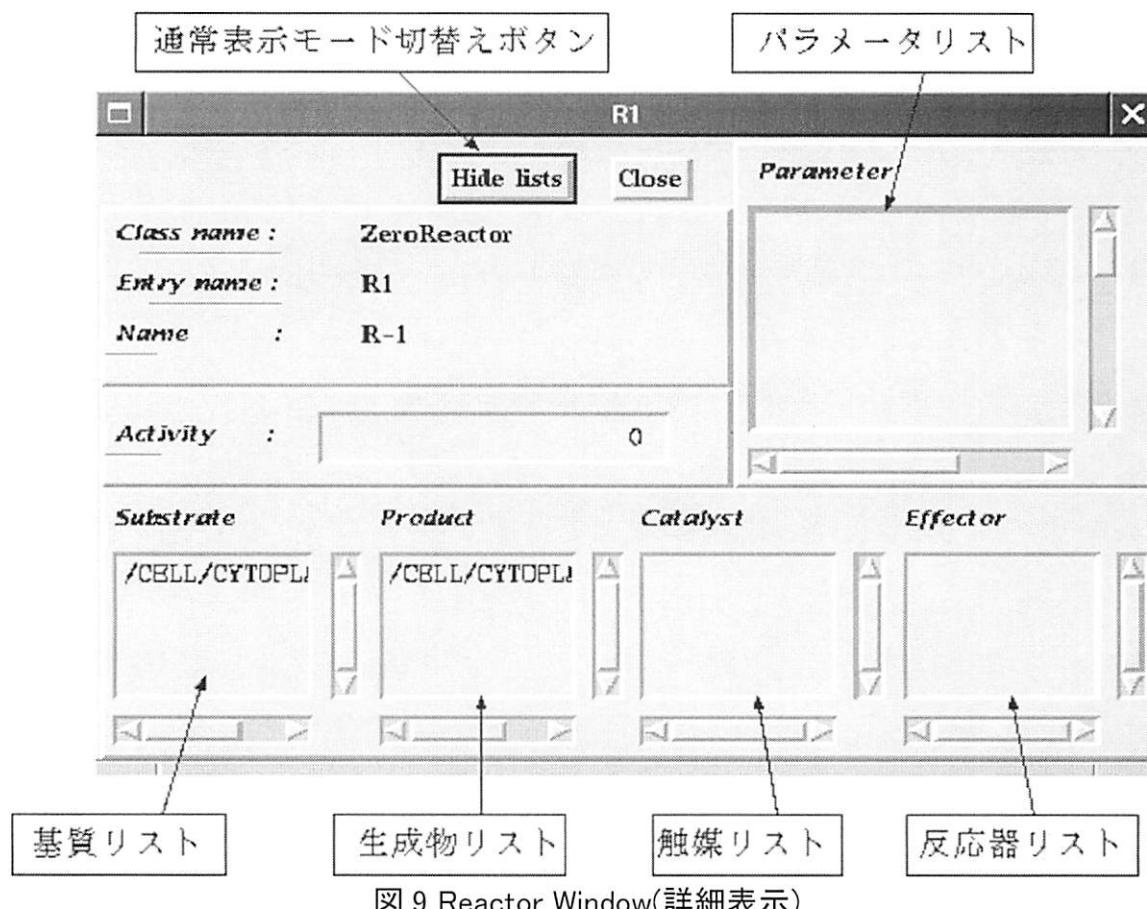


図.9 Reactor Window(詳細表示)

通常表示モード切り替えボタン  
表示を通常表示に切り替える。  
反応物質表示部  
右にパラメータのリストを、また、下に左より基質、生成物、触媒、反応器のリストを表示する。

## 4 E-CELLスクリプトファイル

### 4.1 スクリプトファイルとは

スクリプトファイルは、E-CELL Systemでシミュレーションを行う際に、その開始や停止、結果の記録等のシミュレーションの制御や、その結果を表示する各インターフェイスの制御を一括して行うためのものである。

スクリプトファイルはテキストファイルであり、通常のエディタを用いて作成できる。

### 4.2 スクリプトファイルの文法

スクリプトファイルは上から一行ずつ順番に読み込まれる。ReactorPathおよびStepIntervalは

LoadRuleより先に書かれる必要がある。スクリプトファイルではNewInterface命令をのぞいて、一つの命令は一行に記述され、命令の中途で改行を挟むことは許されない。AddTrace等のNewInterface内で宣言される命令は、それが属するNewInterfaceの命令の直下に、行頭にスペース、またはタブを空けて記述する。

また、「#」記号から行末まではコメントであり、無視される。

E-Cellに自動的に読み込ませる場合には、Ruleより先にReactorを指定する必要がある。

### 4.3 スクリプトファイルのサンプル

スクリプトファイルのサンプルを以下に示す。

```

ReactorPath ../reactors
TmpDir /tmp
LoadRule sample.eri

NewInterface Tracer trace1
Geometry 300x400+20-30
AddTrace Substance:/CELL/CYTOPLASM:A
AddTrace Reactor:/CELL/CYTOPLASM:R1
SaveAt 45

NewInterface SubstanceWindow test-sw
Substance /CELL/CYTOPLASM:A

Run 500
Stop

```

### 4.4 スクリプトファイルの解説

1行目は、シミュレーションで使われるReactorのパスを、2行目は一時ファイルの書き込み先を、3行目では読み込むルールファイルをそれぞれ指定している。

5行目から9行目までは、TracerWindowを一つ表示させ、いくつかの設定を行っている。6行目は表示するWindowの表示位置を画面の左上より横に20ドット、右下より縦に30ドットの位置に、また、サイズを横300ドット、縦400ドットに指定している。7行目はAというSubstanceを、8行目はR1というReactorをTracerに追加するよう指定しており、9行目は開始45秒後にTracerの状態を保存するよう指定している。

11行目から12行目にかけては、サブスタンスウィンドウを開いている。

14行目はシミュレーションを500秒間実行するように、また、15行目で、そのシミュレーションを停止するよう指定している。

### 4.5 スクリプトファイルの命令の一覧

命令	内容
Accumulator クラス名	省略時アキュムレータークラスを指定する。
ReactorPath <i>directory</i>	引数に、そのシミュレーションで使用されるReactorの 実行ファイルのあるディレクトリのパスを指定する。パスは相対パスを用いることも可能である。

<code>TmpDir directory</code>	引数に指定されたパスに、シミュレーションの際に発生した一時ファイルを書き込む。デフォルトでは <code>/var/tmp</code> が指定されている。
<code>LoadRule filename</code>	引数に指定されたルールファイルを読み込む。
<code>StepInterval second</code>	積分時間間隔を秒単位で指定する。省略値は0.001である。
<code>UpdateInterval count</code>	ユーザーインターフェース更新間隔を自然数で指定する。省略値は100で、100回積分するごとに1回ユーザーインターフェースが更新される。
<code>SaveCellState file_name</code>	シミュレーションの状態をファイルに保存する。
<code>LoadCellState file_name</code>	ファイルに保存されているシミュレーションの状態を読み込む。
<code>Run seconds</code>	シミュレーションを引数に指定された(シミュレーション系内の時間で)秒数間実行する。
<code>Stop</code>	シミュレーションを停止する。引数はない。
<code>Exit</code>	E-CELLを終了する。引数はない。
<code>NewInterface interface, name</code>	新しいインターフェイスを作成する命令である。引数を2つとり、第1引数には呼び出すインターフェイスの種類を、第2引数にそのウインドウの名前を指定する。

### NewInterface内で宣言される命令

命令	内容
	そのインターフェイスのサイズと表示する座標を指定する。しかし引数は1つしかとらないため、以下のように記述する。

$(X1)x(Y1)+/- (X2)+/- (Y2)$

<code>Geometory (X1)x(Y1)+/- (X2)+/- (Y2)</code>	ただし、 $(X1)$ 、 $(Y1)$ はそれぞれ横のサイズと、縦のサイズ、また、 $(X2)$ 、 $(Y2)$ はそれぞれ表示されるウインドウの左上の頂点のx座標、y座標を示す。 $+$ の時は画面の左上が基準となり $-$ の時は右下が基準となる。 ただし、ウインドウマネージャによってはこの機能が使えないことがある。
--	--

### 各インターフェイスごとの命令

- Tracer

命令	内容
<code>AddTrace FQPN</code>	Tracer内でのみ宣言される。引数にかかれたSubstance、もしくはReactorをTracerの表示に加える。引数は、IDだけでなくそれが存在するシステムとTypeを含めて示す必要があり、FQPNを用いる。
<code>SaveAt seconds directory</code>	Tracer内でのみ宣言される。ある時点でのTracerの状態を保存し、TracerWindow上でSaveボタンをクリックするのと同じ効果がえられる。第1引数にはどの時点で状態を記録するかを、秒数で与える。第2引数で、データを保存するディレクトリを指定でき、省略すると「./Data」になる。

- SubstanceWindow

命令	内容
<code>Substance FQEN</code>	SubstanceWindow内でのみ宣言される。引数のSubstanceをSubstanceWindowに表示する。引数は、SubstanceのIDだけでなく、

その存在するシステムを含めて示す必要があり、FQENを用いる。

- ReactorWindow

命令

Reactor *FQEN*

内容

ReactorWindow内でのみ宣言される。引数のReactorを

ReactorWindowに表示する。引数はSubstance命令と同様、Reactorの  
IDだけでなく、その存在するシステムを含めて示す必要があり、  
FQENを用いる。



E-CELL User's Manual

Last Update \$Date: 2002/03/13 17:27:41

\$.

1章 基本操作 [ 2章 : 3章 ] [ トップ ]



## INDEX

## 1 はじめに

## 2 スプレッドシートの記述

### 2.1 見出し語について

### 2.2 Systemパート

FQEN,FQPN

### 2.3 Substanceパート

Fix機能

### 2.4 Reactorパート

### 2.5 Includeパート

## 3 スプレッドシートの保存と変換

### 3.1 スプレッドシートの保存

### 3.2 erファイルへの変換

## 1 はじめに

ルールファイルとは、目的のシミュレーションをECELLで実行するための情報を入力するデータファイルを指す。このファイルには、酵素なども含む全ての物質が記載され、これらの物質の相互関係(反応)や、体積などの環境設定が記述されている。さらに、計算に用いる積分方式の指定もルールファイルで行うことができる。ユーザーが定義できるファイルには、ルールファイルの他にReactorがあり、これは第3章で詳しく述べる。Reactorは反応速度式をもったプログラムで、ルールファイルの指定により物質などの量をその式に従って変化させる。

ルールファイルには、

- ssファイル
- erファイル
- eriファイル

の3つがあり、ルールファイルの作成はスプレッドシートに書き込む方法か、erファイルを直接書く方法のどちらかで記述できる。

スプレッドシートによるルールファイルの作成には汎用のスプレッドシートソフトウェアを用いるため、視覚的に理解しやすく、比較的簡単に記述できることが利点である。一方、erファイルはテキストファイルとして記述されるため、マクロを使って簡単に操作することが可能である。これらはいずれも、最終的にE-CELL Systemに読み込まれる形であるeriファイルに変換される(スプレッドシートの場合一度、erファイルに変換されてからeriファイルに変換される)。

ルールファイルはシミュレーションを行うモデル環境に関する定義を行うSystemパートと、物質に関する定義を行うSubstanceパート、反応や相互作用に関する定義を行うReactorパート、他のルールファイル(erファイル)を読み込む際に用いるIncludeパートからなっている。これらはひとまとめに記述する必要はなく、ユーザーがわかりやすい任意の順序で記述することができる。この章ではスプレッドシートを用いたルールファイルの作成について解説する(erファイルの仕様については、章末を参照)。

## 2 スプレッドシートの記述

ここではルールファイルの記述について実例を示しながら、見出し行と呼ばれるパートを区分する行と4つのパートを解説する。  
なお、スプレッドシートの中で、

- タブ
- クオーテーション
- ダブルクオーテーション

を用いることは禁止されている。  
これらの記号は項目を分ける際にスプレッドシートソフトウェアによって使用されるため、これらが項目中に存在すると適正なeriファイルが作成できない。

### 2.1 見出し語について

スプレッドシートでは、記述するパートごとに見出しをつける必要がある。これらの見出し語が入っている行をそれぞれのパートの「見出し行」とよび、パートが変わることごとに各パートの先頭行として記述されなければならない。図で、青く色付けされた行が見出し行である。見出し語は、大文字と小文字の区別をせず、見出し語がどの順序で並んでいてもかまわない。また、各見出し行の最後の列にはmemoという見出し語をつけたことができ、memoの列に書かれた情報はeriファイルに出力されることはない。

Type	Class	path	ID	Name
System	Cell	/	CELL	The cell
System	Environment	/	ENVIRONMENT	The culture medium
System	Cytoplasm	/CELL	CYTOPLASM	The cytoplasm
System	Membrane	/CELL	MEMBRANE	The membrane
Type		path	ID	Name
Substance		/CELL/CYTOPLASM	SA	Substance A
Substance		/CELL/CYTOPLASM	SB	Substance B
Substance		/CELL/CYTOPLASM	SC	Substance C
Substance		/CELL/CYTOPLASM	SD	Substance D
Substance		/CELL/CYTOPLASM	SE	Substance E
Substance		/CELL/CYTOPLASM	E.ab	Isomerase of A
Substance		/CELL/CYTOPLASM	E.bc	Dehydratase of B
Substance		/CELL/CYTOPLASM	E.cd	Isomerase of C
Substance		/CELL/CYTOPLASM	E.de	Isomerase of D
Substance		/CELL/CYTOPLASM	C.Ebc-D	Complex of E.bc and D
Type	Class	path	ID	Name
Reactor	ConstantParameterReactor	/ENVIRONMENT	VOLUME	Volume index for environment
Reactor	ConstantParameterReactor	/CELL/CYTOPLASM	VOLUME	Volume index for cytoplasm
Reactor	MichaelisUniUniReversibleReactor	/CELL/CYTOPLASM	E.ab-0	Isomerization of A
Reactor	MichaelisUniUniReversibleReactor	/CELL/CYTOPLASM	E.bc-0	Dehydration of B
Reactor	MichaelisUniUniReversibleReactor	/CELL/CYTOPLASM	E.cd-0	Isomerization of C
Reactor	MichaelisUniUniReactor	/CELL/CYTOPLASM	E.de-0	Isomerization of D
Reactor	RapidEquilibriumPReactor	/CELL/CYTOPLASM	IEQ-Ebc-D	Bonding of Ebc and D
Type	Filename			
Include	simple.er			

図1 見出し語

## 2.2 Systemパート

シミュレーションを行うには、シミュレーションシステムの構造と環境の設定をしなければならない。それを行うのがSystemパートである。シミュレーションが行われる場所(System)やその体積の定義をSystemパートで行う。

見出し語は下表の通りである。

見出し語	意味	省略時のデフォルト
Type	[System]	省略不可
Class	シミュレーションを行うSystemの種類を	省略不可

**指定**

Path	Systemの階層構造の中でどの場所に位置するかを指定	省略不可
ID	指定したSystemのID	省略不可
Name	指定したSystemの名称	ID
Inside	膜の内側を指定(膜構造を表すSystemでのみ使用)	入力が必要なSystemについて は省略不可
Outside	膜の外側を指定(膜構造を表すSystemでのみ使用)	入力が必要なSystemについて は省略不可
Volumeindex	体積(Volume)を表すReactorを指定	体積(Volume)が必要なSystem では省略不可

表2-1 Systemパート見出し語一覧

Type	Class	path	ID	Name
System	Cell	/	CELL	The cell
System	Environment	/	ENVIRONMENT	The culture medium
System	Cytoplasm	/CELL	CYTOPLASM	The cytoplasm
System	Membrane	/CELL	MEMBRANE	The membrane

図2 Systemパート [Type列、Class列]

Type列には[System]と記入する。Class列では、シミュレーション中で使用するSystem(場所)の種類(Class)を定義する。

Cell, Environment, Cytoplasm, MembraneをClassとしてここで定義することができる。これら全てのClassは、SubstanceやReactor、体積を扱うことができるが、EnvironmentとMembraneについては下位にSystemを作ることができない。各Classの意味は以下の通りである。

	Class	Substance	Reactor	体積	下位のSystem
	Cell	○	○	○	○
	Environment	○	○	○	×
	Cytoplasm	○	○	○	○
	Membrane	○	○	○	×

表2-2 Classの種類

Type	Class	path	ID	Name
System	Cell	/	CELL	The cell
System	Environment	/	ENVIRONMENT	The culture medium
System	Cytoplasm	/CELL	CYTOPLASM	The cytoplasm
System	Membrane	/CELL	MEMBRANE	The membrane

図3 Systemパート [path列]

定義したSystem(場所)において、そのSystemが階層構造のどこにあるかをPath列で指定する。つまり、Systemの住所のようなものである。PathはE-

CELL Systemに組み込まれた最上位のSystemである、RootSystem( / )で始まり、それ以下はユーザによって定義される。Pathは「/system1/system2/system3」のように「/」で区切って表記され、IDの上位Systemまでを全て書き込む。なお、このPathの記載は省略することができない。また、大文字と小文字は区別されるので注意すること。

### FQEN、FQPN

FQEN(FullyQualifiedEntryName)はSystemや、Substance、Reactorなどの階層構造における位置を示すための記述法で、「/system0/system1/system2:ID」のようにPathとIDを「:」を用いて組み合わせる。例えば、Pathが「/CELL/CYTOPLASM」、IDが「ATP」である場合、FQENは「/CELL/CYTOPLASM:ATP」となる。また、FQENではSystemや、Substance、Reactorの区別がつかない。このような情報を含んだ表現がFQPNである。FQPNは、[Type]:[FQEN]で表す。先ほどのATPを用いた例では、「Substance:/CELL/CYTOPLASM:ATP」と表されることになる。これらは全て大文字と小文字が区別されるので注意すること。

Type	Class	path	ID	Name
System	Cell	/	CELL	The cell
System	Environment	/	ENVIRONMENT	The culture medium
System	Cytoplasm	/CELL	CYTOPLASM	The cytoplasm
System	Membrane	/CELL	MEMBRANE	The membrane

図4 Systemパート [ID列、Name列]

IDは、E-CELLシミュレーションソフトウェア内の処理番号に相当する。このため、一つのSystem内に同一のIDが存在することはできない。同じIDをいる場合、Warningメッセージが表示されて先に定義されたIDが採用される。また、IDは大文字、小文字を区別するので注意しなければならない。このID列には定義したSystem(場所)のIDを、Nameには、定義したSystemの名称を記入する。

ID	Name	Inside	Outside	VolumeInc
CELL	The cell			
ENVIRONMENT	The culture medium			/ENVIRON
CYTOPLASM	The cytoplasm			/CELL/CY
MEMBRANE	The membrane	/CELL:CYTOPLASM	/ENVIRONMENT	

図5 Systemパート [Inside列、Outside列]

inside/outside列は、その行で定義しているClass(Systemの種類)がMembrane(膜)であるとき、膜の内側もしくは外側にあるSystemを示すのに用いる。

現在の E-CELL Systemではこの情報を用いた処理を行っていないが、今後、アクセスできる System を限定したりする処理を加える予定である。例では、MEMBRANEというSystemがCYTOPLASMとENVIRONMENTに挟まれた膜であることを示している。

Inside	Outside	VolumeIndex	Memo
/CELL:CYTOPLASM	/ENVIRONMENT	/ENVIRONMENT:VOLUME	Definition of CellComponent
		/CELL/CYTOPLASM:VOLUME	Definition of CellComponent
			Definition of CellComponent

図6 Systemパート [VolumeIndex列]

E-CELL System では、Systemの体積を Reactorを用いて表現する。System の VolumenIndex として Reactorの FQEN を指定すると、その Reactorの activity の値が体積(単位:[L])として解釈される。System の VolumenIndex を指定した場合、その FQEN に相当する Reactorを Reactorパートで定義しなければならない。Systemは体積を指定しない場合もありうる。例えば、上図のように Membrane は体積を定義していない。ただし、Reactorが物質の濃度を用いて計算を行なう場合や、Substance パートで物質の濃度(Conc)を指定する場合、濃度を分子数に変換する際に体積の値が必要である。これらの場合にはその物質が存在している Systemの VolumeIndex は必ず指定する必要がある。

### 2.3 Substanceパート

シミュレーションに関わる全ての物質(Substance)は、ルールファイル内で定義される必要がある。それらの物質の定義を記述するのがSubstanceパートである。

Substanceパートの見出し語は下表の通りである。

見出し語	意味	省略時のデフォルト
Type	[Substance]	省略不可
Class	通常は[Substance]。省略してもよい。	省略不可
Path	Substanceのある場所(System)	省略不可
ID	指定したSubstanceのID	省略不可
Name	指定したSubstanceの名称	ID
Qty	Substanceの個数(分子数)の初期値(Concとは両立しない)	0
Conc	Substanceの濃度M(mol/l)の初期値(Qtyとは両立しない)	Qty
Arg_tag	Substanceの小数部分の累積方法を指定したい場合 [Accumulator]	省略可
Arg_coeff	Substanceの小数部分の累積方法を選択	Arg_tag がある場合 省略不可

表2-3 Substanceパート見出し語一覧

Type	Class	path	ID	Name
Substance	Substance	/CELL/CYTOPLASM	SA	Substance A
Substance	Substance	/CELL/CYTOPLASM	SB	Substance B
Substance	Substance	/CELL/CYTOPLASM	SC	Substance C
Substance	Substance	/CELL/CYTOPLASM	SD	Substance D
Substance	Substance	/CELL/CYTOPLASM	SE	Substance E
Substance	Substance	/CELL/CYTOPLASM	E.ab	Isomerase of A
Type	Class	path	ID	Name
Substance	Substance	/CELL/CYTOPLASM	E.bc	Dehydratase of B
Substance	Substance	/CELL/CYTOPLASM	E.cd	Isomerase of C
Substance	Substance	/CELL/CYTOPLASM	E.de	Isomerase of D
Substance	Substance	/CELL/CYTOPLASM	C.Ebc-D	Complex of E.bc a

図7 Substanceパート [Type列、Class列]

Type列には「Substance」と入れる。Class列では、各行に入っているものの型を記入する。Substanceパートでは、Class(種類)として指定できるのは今のところSubstanceのみとなっている。従って、Class列にはSubstanceと入力する。ここを省略すると、Substanceが自動的に入力される。

Type	Class	path	ID	Name
Substance	Substance	/CELL/CYTOPLASM	SA	Substance A
Substance	Substance	/CELL/CYTOPLASM	SB	Substance B
Substance	Substance	/CELL/CYTOPLASM	SC	Substance C
Substance	Substance	/CELL/CYTOPLASM	SD	Substance D
Substance	Substance	/CELL/CYTOPLASM	SE	Substance E
Substance	Substance	/CELL/CYTOPLASM	E.ab	Isomerase of A
Type	Class	path	ID	Name
Substance	Substance	/CELL/CYTOPLASM	E.bc	Dehydratase of B
Substance	Substance	/CELL/CYTOPLASM	E.cd	Isomerase of C
Substance	Substance	/CELL/CYTOPLASM	E.de	Isomerase of D
Substance	Substance	/CELL/CYTOPLASM	C.Ebc-D	Complex of E.bc a

図8 Substanceパート [path列]

Path列では、物質が存在する場所(System)を指定する。Pathの記述法はSystemパートのPath列と同様に「/system1/system2/system3」のように「/」で区切って入力する。なお、同じID,Nameを持つ物質でも、異なるSystemを指定していればE-CELL Systemでは違うSubstanceとして認識されるため、注意が必要である。大文字と小文字は区別されるので注意すること。

Type	Class	path	ID	Name
Substance	Substance	/CELL/CYTOPLASM	SA	Substance A
Substance	Substance	/CELL/CYTOPLASM	SB	Substance B
Substance	Substance	/CELL/CYTOPLASM	SC	Substance C
Substance	Substance	/CELL/CYTOPLASM	SD	Substance D
Substance	Substance	/CELL/CYTOPLASM	SE	Substance E
Substance	Substance	/CELL/CYTOPLASM	E.ab	Isomerase of A
Type	Class	path	ID	Name
Substance	Substance	/CELL/CYTOPLASM	E.bc	Dehydratase of E
Substance	Substance	/CELL/CYTOPLASM	E.cd	Isomerase of C
Substance	Substance	/CELL/CYTOPLASM	E.de	Isomerase of D
Substance	Substance	/CELL/CYTOPLASM	C.Ebc-D	Complex of E.bc a

図9 Substanceパート [ID列、Name列]

SubstanceのIDとNameを記入する。IDでは、大文字と小文字の区別が行われるので注意が必要である。

path	ID	Name	QTY	CONC
/CELL/CYTOPLASM	SA	Substance A		1000000
			Fix	
/CELL/CYTOPLASM	SB	Substance B		0
/CELL/CYTOPLASM	SC	Substance C		0
/CELL/CYTOPLASM	SD	Substance D		0
/CELL/CYTOPLASM	SE	Substance E		0
/CELL/CYTOPLASM	E.ab	Isomerase of A		
			Fix	
/CELL/CYTOPLASM	E.bc	Dehydratase of B		
/CELL/CYTOPLASM	E.cd	Isomerase of C		
/CELL/CYTOPLASM	E.de	Isomerase of D		
/CELL/CYTOPLASM	C.Ebc-D	Complex of E.bc and D		

図10 Substanceパート [QTY列]

Qtyは、ユーザーが定義した体積における物質の個数(分子数)の初期値を表している。Concとは両立せず、両方記述されている場合はWarningメッセージが出てQtyの値が採用される。また、どちらの数値も入力されていない時は、Qty=0として計算される。また、そのQtyの値を反応によって増減させる必要がないとき、Fix機能によって値を固定することができる(Fix機能については後の項を参照)。

ID	Name	QTY	CONC	Memo
SA	Substance A	1000000		1e+06 means SA is fixed
		Fix		
SB	Substance B	0		
SC	Substance C	0		
SD	Substance D	0		
SE	Substance E	0		
E.ab	Isomerase of A		0.83	
		Fix		
E.bc	Dehydratase of B		0.02	Accumulator
E.cd	Isomerase of C		0.01	
E.de	Isomerase of D		0.01	
C.Ebc-D	Complex of E.bc and D		0	

図11 Substanceパート [Conc列]

ConcにはM (mol/l)を単位とした物質の濃度を入力する。eriファイルに変換する際に、Systemパートで指定した体積を表すReactorのint\_actで指定した体積とE-CELL System内のアボガドロ定数(6.0221367e23)で分子数に変換される。Qtyとは両立せず、Concに何も入っていない場合は、Qtyの値がeriファイルに出力される。また、Conc値を記載したにも関わらず、Systemパートで物質が存在するシステムの体積の定義(VolumeIndex)がない場合はVolume=0で処理されるので注意が必要である。ここでも、Fix機能を使って値を固定することが可能である。

## Fix機能

ID	Name	QTY	CONC	Memo
SA	Substance A	1000000		1e+06 means SA is fixed
SB	Substance B	Fix	0	
SC	Substance C		0	
SD	Substance D		0	
SE	Substance E		0	
E.ab	Isomerase of A		0.63	
E.bc	Dehydratase of B	Fix		0.02 Accumulator
E.cd	Isomerase of C			0.01
E.de	Isomerase of D			0.01
C.Ebc-D	Complex of E.bc and D			0

図12 Fix機能

Fix機能とは、指定された Substance 行の物質を Reactor による増減の影響を受けないようにして、入力した Qty または Conc の値を保つ機能である。Fix 指定するには、維持したい値が記載してある Qty、または、Conc のすぐ下の行に「Fix」と記入する。

ID	Name	CONC	Arg_tag	Arg_coeff
E.bc	Dehydratase of B	2.00E-002	Accumulator	SimpleAcc
E.cd	Isomerase of C	0.01		
E.de	Isomerase of D	0.01		
C.Ebc-D	Complex of E.bc and D	0		

図13 Substanceパート [Arg\_tag列]

Accumulatorとは、小数部分の累積方法を表現したものである。Arg\_tag に「Accumulator」と記入することにより、Substanceの小数部分の累積方法を指定することが可能になる。累積方法の具体的な選択肢は次の Arg\_coeff の節で挙げる。ただし、各物質ごとに異なる累積方法を指定する場合、計算精度は未保証である。

Name	CONC	Arg_tag	Arg_coeff
Dehydratase of B	2.00E-002	Accumulator	SimpleAccumulator
Isomerase of C	0.01		
Isomerase of D	0.01		
Complex of E.bc and D	0		

図14 Substanceパート [Arg\_coeff列]

Arg\_coeff では、Substance の累積方法をカプセル化した Accumulator クラスを選択して記入する。以下に、その選択肢を示す。

## Accumulator クラス

## 小数部分の累積方法

整数部分と小数部分を別々に記憶する。値を使う場合には、整数部分だけが使われる。値を更新する場合には、変化量の小数部分を積み立て、積み立て量が1.0以上になると、整数部に繰り上げる。例を示す。

## ReserveAccumulator

## 変化量 整数部分 小数部分

初期値	0	0
0.8	0	0.8
0.8	1	0.6
0.8	2	0.4

## SimpleAccumulator

実数をそのままの値で計算する。精度はi386版で10進18桁、alpha版で10進15桁である。

## RoundDownAccumulator

小数点以下を切り捨て計算する。

## RoundOffAccumulator

小数点第一位を四捨五入して計算する。

## MonteCarloAccumulator

整数部分と小数部分を分け、小数部分を確率として乱数で切り上げまたは切り捨てる。例えば、変化量が3.4ならば、0.6の確率で3変化し、0.4の確率で4変化する。

## 2.4 Reactorパート

Substanceパートで定義した物質を用いた反応や、物質間相互作用、さらに体積などの環境変数を扱う式の指定を行う部分がReactorパートである。Reacterパートでは、反応速度式(Reacter)を指定し、反応に関わる Substance を指定して、それらの化学量論的記述を行う。  
Reacter行の見出し語は下表の通りである。

見出し語	意味	省略時のデフォルト
Type	[Reactor]	省略不可
Class	Reactor(反応速度式)を指定	省略不可
Path	反応が起こる場所(System)	省略不可
ID	反応のID	省略不可
Name	反応の名称	ID
init_act	ReactorActivityの初期値	特別な場合(VolumeIndex に指定した場合など)を除き省略可
S_ID	反応基質(Substrate)のID	省略不可
S_Path	Substrateの存在している場所(System)	Path
S_Coeff	Substrateの化学量論的係数	1
P_ID	反応生成物(Product)のID	省略不可
P_Path	Productの存在している場所(System)	Path
P_Coeff	Productの化学量論的係数	1
C_ID	反応触媒(Catalyst)のID	省略不可
C_Path	Catalystの存在する場所(System)	Path

arg_tag	Reactorの定数名(反応速度式が使用する定数名)	入力が必要なReactorについては省略不可
arg_coeff	定数名に対応する定数の値。	arg_tagを入力したReactorについては省略不可
E_ID	反応に影響を与える物質(Effecter)	指定する必要がなければ省略可
E_Path	Effecterのある場所(System)	Path
E_Coeff	Effecterのとる係数(内容はReactorごとに異なる)	1

表2-4 Reactor/パート見出し語一覧

Type	Class	path	ID	Name
Reactor	MichaelisUniUniReversibleReactor	/CELL/CYTOPLASM	E.ab-0	Isomerization of A
Reactor	MichaelisUniUniReversibleReactor	/CELL/CYTOPLASM	E.bc-0	Dehydration of B
Reactor	MichaelisUniUniReversibleReactor	/CELL/CYTOPLASM	E.cd-0	Isomerization of C
Reactor	MichaelisUniUniReversibleReactor	/CELL/CYTOPLASM	E.de-0	Isomerization of D
Reactor	RapidEquilibriumPReactor	/CELL/CYTOPLASM	!EQ-Ebc-D	Bonding of Ebc and D
Type	Class	path	ID	Name
Reactor	ConstantParameterReactor	/ENVIRONMENT	VOLUME	Volume index for c
Reactor	ConstantParameterReactor	/CELL/CYTOPLASM	VOLUME	Volume index for c

図15 Reactor/パート [Type列、Class列]

Reactorの種類(Class: 反応速度式)はclass列で指定する。この列では、指定したいReactorを「○○Reactor」という形で記入する。この例では、MichaelisMenten式に基づいた反応速度式を用いているため、「Michaelis○○Reactor」が使われている。例中の下図は、体積をもつReactorを宣言している。その他の標準添付されているReactorがどの反応速度式を表しているかは、巻末のスペックシートに記載してある。また、大文字と小文字は区別されるので区別する。反応、または相互作用のIDをID列に記入する。大文字、小文字は区別されるので注意すること。

Type	Class	path	ID	Name
Reactor	MichaelisUniUniReversibleReactor	/CELL/CYTOPLASM	E.ab-0	Isomerization of A
Reactor	MichaelisUniUniReversibleReactor	/CELL/CYTOPLASM	E.bc-0	Dehydration of B
Reactor	MichaelisUniUniReversibleReactor	/CELL/CYTOPLASM	E.cd-0	Isomerization of C
Reactor	MichaelisUniUniReversibleReactor	/CELL/CYTOPLASM	E.de-0	Isomerization of D
Reactor	RapidEquilibriumPReactor	/CELL/CYTOPLASM	IEQ-Ebc-D	Bonding of Ebc and D
Type	Class	path	ID	Name
Reactor	ConstantParameterReactor	/ENVIRONMENT	VOLUME	Volume index for e
Reactor	ConstantParameterReactor	/CELL/CYTOPLASM	VOLUME	Volume index for c

図16 Reactor/パート [path列]

Path列では反応や相互作用の起こるSystem(場所)のPathを入力する。Pathの入力法はSystem/パートや、SubstanceパートのPath列と同じである。この例では、「/CELL/CYTOPLASM」で反応が起こっていることがわかる。大文字と小文字は区別されるので注意すること。

Class	path	ID	Name
MichaelisUniUniReversibleReactor	/CELL/CYTOPLASM	E.ab-0	Isomerization of A
MichaelisUniUniReversibleReactor	/CELL/CYTOPLASM	E.bc-0	Dehydration of B
MichaelisUniUniReversibleReactor	/CELL/CYTOPLASM	E.cd-0	Isomerization of C
MichaelisUniUniReversibleReactor	/CELL/CYTOPLASM	E.de-0	Isomerization of D
RapidEquilibriumPReactor	/CELL/CYTOPLASM	IEQ-Ebc-D	Bonding of Ebc and D
Type	Class	path	ID
Reactor	ConstantParameterReactor	/ENVIRONMENT	VOLUME
Reactor	ConstantParameterReactor	/CELL/CYTOPLASM	VOLUME

図17 Reactor/パート [ID列、Name列]

Name列には反応または相互作用の名称を記入する。ここになにも記載されていない場合は、IDに記入されている情報がeriファイルに出力される。体積をもつReactor(下2段)については、それが体積であることが分かることをNameをつけておくことが望ましい(図を参照)。また、posterior Reactor

を指定する場合は ID を「!」で始まるものにする必要がある(詳しくは第3章を参照)。ただし FluxReactor 系など、通常の Reactor の ID を「!」で始まるものにした場合、その動作は無保証であるため注意が必要である。

	Arg_tag	Arg_coeff	init_act	Memo
environment	Value		1.00E-015	1.00E-015 Definition of \
cytoplasm	Value		1.00E-018	1.00E-018 Definition of \

図18 Reactorパート [init\_act列]

シミュレーションが1ステップ進むごとに計算される Reactor の activity は、シミュレーション開始時に Reactor の activity が必要な場合は初期 activity を設定しなければならない。init\_act列は、その値を入力するために用意されている。SystemパートでVolumeIndexに記載したReactorを定義する場合、Reactorパートでinit\_act列の値を必ず入力しなければならない。このinit\_actはE-CELL Systemが計算を開始する以前にReactorの activityを参照しなければならないときに初期値として入力される。この値がない場合は、シミュレーションを開始した一番最初のstepで体積を用いる全計算にVolume=0が代入され、結果としてシミュレーションができない。例ではSystemの体積を示すReactorの ConstantParameterReactor での体積の初期値を定義している。

path	ID	Name	S_ID	S_path	S_Coeff
/CELL/CYTOPLASM	E.ab-0	Isomerization of A	SA	/CELL/CYTOPLASM	
/CELL/CYTOPLASM	E.bc-0	Dehydration of B	SB	/CELL/CYTOPLASM	
/CELL/CYTOPLASM	E.cd-0	Isomerization of C	SC	/CELL/CYTOPLASM	
/CELL/CYTOPLASM	E.de-0	Isomerization of D	SD	/CELL/CYTOPLASM	
/CELL/CYTOPLASM	!EQ-Ebc-D	Bonding of Ebc and D	E.bc	/CELL/CYTOPLASM	
			SD	/CELL/CYTOPLASM	

図19 Reactorパート [S\_ID列、S\_path列、S\_Coeff列]

S\_IDではSubstrate(基質)のIDをS\_PathではSubstrateが存在している System(場所)のPathを指定する。この反応では、CYTOPLASMに存在しているATPとGlcがSubstrateとして反応に関わっている。S\_Coeffには、 Substrateの化学量論的係数を記入する。

Name	S_ID	S_path	S_Coeff	P_ID	P_path
Isomerization of A	SA	/CELL/CYTOPLASM	1 SB		/CELL/CYTOPLASM
Dehydration of B	SB	/CELL/CYTOPLASM	1 SC		/CELL/CYTOPLASM
Isomerization of C	SC	/CELL/CYTOPLASM	1 SD		/CELL/CYTOPLASM
Isomerization of D	SD	/CELL/CYTOPLASM	1 SE		/CELL/CYTOPLASM
Bonding of Ebc and D	E.bc SD	/CELL/CYTOPLASM /CELL/CYTOPLASM	1 C.Ebc-D 1		/CELL/CYTOPLASM

図20 Reactor/パート [P\_ID列、P\_path列、P\_Coeff列]

P\_IDにはProduct(生成物)となる物質のIDを、P\_PathではProductが存在しているSystem(場所)のPathを指定する。ここでは、ProductとなるADPとG6PがCYTOPLASMに存在していることを示している。P\_Coeffには、Productの化学量論的係数を入力する。

S_path	S_Coeff	P_ID	P_path	P_Coeff	C_ID	C_path
/CELL/CYTOPLASM	1 SB		/CELL/CYTOPLASM	1 E.ab		/CELL/CYTOPL
/CELL/CYTOPLASM	1 SC		/CELL/CYTOPLASM	1 E.bc		/CELL/CYTOPL
/CELL/CYTOPLASM	1 SD		/CELL/CYTOPLASM	1 E.cd		/CELL/CYTOPL
/CELL/CYTOPLASM	1 SE		/CELL/CYTOPLASM	1 E.de		/CELL/CYTOPL
/CELL/CYTOPLASM	1 C.Ebc-D		/CELL/CYTOPLASM	1		
/CELL/CYTOPLASM	1					

図21 Reactor/パート [C\_ID列、C\_path列]

C\_IDには反応において実際にSubstarateやProductに対してCatalyst(触媒)となって作用するもののIDを記入する。C\_PathではCatalystの存在するSystem(場所)のPathを指定する。通常、このCatalystの ID, Path は、特に理由がない限りは反応の ID, Path と同じになる。

	S_Coeff	P_ID	P_path	P_Coeff	C_ID	C_path	Arg_f
PLASM	1 SB		/CELL/CYTOPLASM		1 E.ab	/CELL/CYTOPLASM	KmS KmP KcF KcR
PLASM	1 SC		/CELL/CYTOPLASM		1 E.bc	/CELL/CYTOPLASM	KmS KmP KcF KcR
PLASM	1 SD		/CELL/CYTOPLASM		1 E.cd	/CELL/CYTOPLASM	KmS KmP KcF KcR
PLASM	1 SE		/CELL/CYTOPLASM		1 E.de	/CELL/CYTOPLASM	KmS KcF KcR
PLASM	1 C.Ebc-D		/CELL/CYTOPLASM		1		Keq
PLASM	1						

ID	Name	Arg_tag	Arg_coeff	init_act
VOLUME	Volume index for environment	Value		1.00E-015
VOLUME	Volume index for cytoplasm	Value		1.00E-018

図22 Reactorパート [Arg\_tag列、Arg\_coeff列]

arg\_tag列ではReactorで使用する定数名、つまり反応速度式が使用する定数を入力する。この定数名は、Reactorごとに決まっているものである、上の図、下から2つめのReactor例では、MichaelisUniUniReactorを反応速度式としているので、arg\_tagは「KcF,KmS」になっている(標準添付されているReactorがどのような定数を必要としているかは巻末のスペックシートを参照)。複数ある場合は、次の行に記述し、この場合、追加した行には他の項目について記述する必要はない(図を参照)。また、この列も大文字、小文字が区別されるので注意すること。体積を表すために用いる標準添付のConstantParameterReactor(体積計算用のReactor)の場合は、ここに「Value」と記入する(例中、下の図)。

arg\_tagのすぐとなりにarg\_coeffを記入する。arg\_coeffには、arg\_tagに対応する定数の値が入り、さきほどの例ではMichaelisUniUniReactorが使用する「KcF,KmS」の値が入っている。体積を表すために用いる標準添付のConstantParameterReactorの場合は、ここにSystemの体積を記入する(図参照)。E-CELL Systemでは最初の1ステップ(1計算時間幅)のみinit\_actに記載された体積で計算され、以後のステップはReactorの示す値で計算される(ConstantParameterReactorではValueの値)。

このほか、SubstrateやProductのようにReactorによる量的な増減は受けないが、反応速度に関わってこれに影響を与えるものをEffectorとして定義することができる。E\_IDにはEffectorとして指定したいSubstanceのIDを、E\_PathではそのEffectorの存在するSystem(場所)をPathで指定する。E\_CoeffはEffectorの係数を表すが、実際に入力されるべき内容はReactorごとに自由に決められるので、そのReactorの仕様書を参照すること。

## 2.5 Includeパート

Reactor	RapidEquilibriumPReactor	/CELL/CYTOPLASM !EQ-Ebc-D	Bonding of Ebc and
Type	Filename		
Include	simple.er		

図23 Includeパート

スプレッドシートにIncludeパートを記述することで、指定したerファイルを挿入することが可能である。このIncludeパートは、他のパート同様、ルールファイル中のどの場所にいくつあってもかまわない。いくつかのルールファイルに共通した行などを別のerファイルとして記述し、出力されるeriファイル(ECELLに読み込まれる最終形式のデータファイル)にその記述を組み込みたい場合などに有用である。シミュレーションに必要な全ての情報をスプレッドシートに書き込んでおり、他のファイルの読み込みを必要としない場合は、このIncludeパートの記述は省略する。Includeパートでその別ファイルを指定すれば、その情報はeriファイルに組み込まれて出力される。Includeパートの見出し語はType,Filenameである。Typeには[Include]と記入し、Filenameには読み込みたいファイル名を指定する。例では「sample.er」というファイルを読み込んでいる。

### 3 スプレッドシートの保存と変換

#### 3.1 スプレッドシートの保存

E-CELL Systemに読み込まれるルールファイルフォーマットであるeriファイルに変換するには、変換ソフトウェアが認識することのできる形であるTAB区切りのテキストファイルで保存する必要がある。

#### 3.2 erファイルへの変換

E-CELL Systemでシミュレーションを行うには、ss2erというスプレッドシートをerファイルへ変換するソフトウェアと、erファイルを実行形式のeriファイルへ変換するer2eriという変換ソフトウェアの2つを経てeriファイルに変換する。スプレッドシートはss2erが読み込めるTAB区切りのテキストファイルでセーブされ、次いで、コマンドラインにおいて

```
% ss2er < filename.tex > filename.er
```

と入力して変換する。これにより、スプレッドシートファイルからerファイルが生成される。

erファイルを直接書いている場合は上述の作業は省略する(erファイルについては次章を参照)。次に、erファイルをE-CELL Systemが読み込む最終形式のファイルであるeriファイルに変換する。

変換は、コマンドラインにおいて

```
% er2eri filename.er > filename.eri
```

と入力することによって実行される。



E-CELL User's Manual

2章 ルールファイルの作成 [ 1章 : 3章 ][ トップ ]



## INDEX

**1.Reactorの概要****2.Reactor Descriptionファイル****2.1 Reactor Descriptionファイルの書き方**

一般情報に関するキーワード

Reactor Spec Sheetに関するキーワード

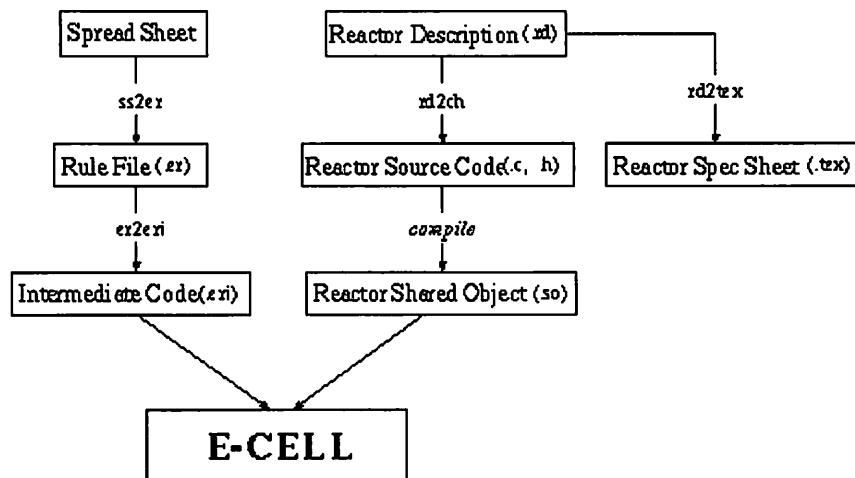
Reactor Source Codeに関するキーワード

**2.2 Reactor Descriptionの例****3 Reactor Source Code(.C .hファイル)への変換****4 Spec Sheetへの変換****5 Makefileを用いたReactor Shared ObjectとSpec Sheetの自動作成****6 ecellの実行****7 E-CELL projectが標準で用意するReactor**

# 1 Reactorの概要

E-CELL Systemは、細胞を、物質を表すSubstance、反応を表すReactor及びそれらが存在する場所、あるいは機能を表現するSystemの三種類のオブジェクトの組合せによりモデル化する。ReactorはSubstanceの量の時間的変化を表現しており、E-CELL Systemを用いたモデリングにおいて、重要な役割を担っている。

Reactorはユーザーが自由にプログラムできるオブジェクトである。ReactorはE-CELL System本体とは別にコンパイルし、E-CELL Systemがロード可能な形式に変換する。E-CELL Systemは、ルールファイルによって指定されたReactorをロードする。Reactorは、C++言語を用いてユーザーが自由に記述できるため、化学反応のほかどのような物質の量の変化の形式も表現できる。



上図は、E-CELL Systemに関するファイルの作成の流れを表している。上図の構造を見ると分かるように、ReactorはE-CELL System本体と別に作成できる。Reactorを作成するには、まず、Reactorの仕様と実行される処理の内容を Reactor Descriptionファイル(RDファイル)に書く。次に RDファイルはC++のコードに変換されさらに E-CELL Systemがロードできる形式(.soファイル)にコンパイルされる。また、RDファイルからLaTeX形式のReactor Spec Sheetに変換できる。

## 裏口(Postern) リアクター

通常(Normal)と裏口(Postern)の2種類のリアクターが存在する。通常リアクターは、積分機構を使って連立微分方程式を解くために使われる。裏口リアクターは、積分機構から独立しており、Substanceの量を計算するために連立微分方程式が適さない場合に使われる。2種類のリアクターは、複数段階の計算機構によって実現されている。まず、通常リアクターによる計算がReact段階で実行され、次に、裏口リアクターによる計算がPosterior段階で実行される。裏口リアクターは次のような状況で利用可能である。

- サブスタンスの量が離散的に制御される場合、例えば、サブスタンスの量が極めて少ない場合。
- サブスタンスの量が、積分機構によらずに、直接決定される場合。例えば、リアクターが転写の開始を決定する場合や、細胞分裂によって全てのサブスタンス量が2分の1になる場合。
- 代数方程式によって反応が計算される場合。例えば、リアクターが迅速平衡連立方程式を解く場合や、浸透圧を計算する場合。

通常リアクターがサブスタンス量を決定するために反応速度を計算するのに対し、裏口リアクターは運動問題を考慮せずに全てのサブスタンスを扱う。したがって、通常リアクターはvelocity(Float v)メソッドを使って計算結果を返すのに対し、裏口リアクターはSetQuantity(Float Q)メソッドを使って計算結果を返すよう実装される。

裏口リアクターは、通常``PReactor''で終わる名前、例えば``ABCPReactor''のような名前を持つ。

それぞれの裏口リアクターによって、1個のみのサブスタンス量が操作されるべきである。複数の裏口リアクターによってはサブスタンス量を制御できない。1個のサブスタンス量の決定に複数の裏口リアクターが関係すると、シミュレーションの精度が保証されない。複数の反応が1つのサブスタンスに影響する場合には、それらの全ての反応を1つの裏口リアクター、例えばGeneralRapidEquilibriumPReactorによって指定する必要がある。

## 2 Reactor Descriptionファイル

いくつかの反応形式のためのReactorは E-CELL Systemに標準添付されている(7節を参照)。しかし、非標準的な反応形式をモデル化する際や、より高度なシミュレーションをするためにはRDファイルを書いて、自分でReactorを作成する必要がある。

### 2.1 Reactor Descriptionファイルの書き方

RDファイルは、キーワードと値の組からなる行を必要な数を並べて作成する。なお、RDファイルを作成する際には以下の約束事に注意する。

- キーワードはアルファベットの大文字又は\_で構成され、スペースを含まない。
- キーワードは @ または % で始まる。@ で始まる行は単純にその内容が読み込まれる。% で始まる行はその中身を，“”で区切ることによって、配列として処理される。
- # で始まる行はコメント文とみなされる。行頭の # をそのまま出力する際は、# とするかスペースを1つ入れる。ただし、行頭以外の # はそのまま出力される。
- 行頭にキーワードのない行は、それより前の行の内容に続くものとして解釈される。(Spec Sheet に変換する際に改行したい場合、改行したい個所に#を入れる。)
- 実際にRDファイルを記述する際は、予めキーワードが一通り書かれて用意されているdefault.rdというファイルに、必要事項を書き込んでいけばいいようになっている。
- キーワードは必要でないものに関しては省略できる。(表1参照)

### 一般情報に関するキーワード

- @CLASSNAME:
- @BASECLASS:
- @AUTHOR:
- @EMAIL:
- @DATE:
- %VERSION:
- @BRIEF\_DESCRIPTION:

@CLASSNAME行には、作成するReactor のクラス名を入力する。ファイル名は`CLASSNAME.rd`でなければならぬ。@BASECLASS行では、そのクラスが継承する元の基底クラスを記述する。通常はFluxReactorを継承する場合が多い。@AUTHOR行、@EMAIL行、@DATE行には、それぞれ作成者、E-mail address、作成日を記入する。日付の書式に厳密な決まりはない。%VERSION行には、E-CELL SystemおよびそのReactorバージョン番号を`'`で区切って記述する。@BRIEF\_DESCRIPTION行は、そのReactorの1行程度の説明。これらのキーワードの内容は、E-CELL System本体にも渡される情報である。

## Reactor Spec Sheetに関するキーワード

- @DESCRIPTION:
- @EQUATION:
- %SUBSTANCE:
- @NOTES:

@DESCRIPTION行は、そのReactor の詳しい説明である。

@EQUATION行は、Reactor の式を LaTeXのdisplaymath環境で記述する。displaymath環境にするには、

1. \$\$<数式の記述>\$\$
2. \$\begin{aligned} &\text{\\$begin\{displaymath}\langle数式の記述\rangle\\$end\{displaymath}} \\ &\text{\\$[\langle数式の記述\rangle\\$]} \end{aligned}

のいずれかの方式で記述すること。

%SUBSTANCE行では、Substanceの定義を行う。用意されているSubstanceは、Substrate、Product、Catalyst、Effectorの4種類があるが、それ以外にもユーザーで定義できるようになっている。ここで記述するものは物質等の名前、最大の個数、最小の個数、備考で、それぞれを`'`で区切る。最大の個数及び最小の個数は、基本的には10進整数値であるが、最大の個数に関しては`'Inf'`として無限大に設定可能である。備考欄は、特殊なSubstanceを定義する際に書けば良い。ここでの記述は、現在はSpec Sheetに参照されるだけである。

@NOTES行では、Reactorの実装に関する注意点を書く。

## Reactor Source Codeに関するキーワード

- %PARAMETER:
- %INCLUDE\_FILE\_H:
- @PRIVATE:
- @PROTECTED:
- @PUBLIC:
- %INCLUDE\_FILE\_C:
- @INITIALIZE\_FUNC:
- @REACT\_FUNC:
- @OPTION\_C:

%PARAMETER行ではそのReactorの引数となるパラメータの定義を行う。記述方法は、パラメータ名、パラメータの型、単位、パラメータに関する記述を、`'`で区切って書く。パラメータの型は、整数ならば`'Int'`を、小数ならば`'Float'`を使用する。しかし、これらの型はE-CELL System独自のもので、C++で使われる`'int'`や`'float'`とは異なる。`'Int'`は64bitの整数、`'Float'`はアルファCPUでは64ビット、

386系CPUでは80ビットの浮動小数である。

%INCLUDE\_FILE\_H行は、ソースコードのヘッダファイル(.h)にincludeしたいファイル名を書く。ファイル名を、<>や" "でくくってはならない。

@PRIVATE行に記述した内容は、オブジェクトの私的要素になる。また、@PROTECTED行での記述は、オブジェクトの限定公開要素になる。 @PUBLIC行での記述は、オブジェクトの公開要素になる。

%INCLUDE\_FILE\_C行は、ソースファイル(.c)の頭にincludeしたいファイル名を記入する。ファイル名を、<>や" "でくくってはならない。

FluxReactor.hを継承しているならば、Reactor.h、Reactant.h、Stepper.h、RootSystem.hは自動的にインクルードされるので記述する必要がない。FluxReactor.hを継承しないならば、前述の4個のファイルを自動的にインクルードするために、StandardHeaders.hのインクルードが望ましい。より詳しい情報はパッケージのSerizawaディレクトリーに入っているヘッダーファイルにある。

@OPTION\_C行では、新たなメソッドやマクロの定義等を書く。ここに書いた事項は、.cファイルの先頭に挿入される。

@INITIALIZE\_FUNCにはルールファイルの読み込み後、シミュレーションの開始前に一度だけ実行する処理をC++のコードで記述する。ここでは主にパラメータの値域のチェックや、シミュレーション中に変化しない値の計算などの初期設定を行う。

@REACT\_FUNCには毎ステップ行う処理を書く。通常は、

1. 反応速度を算出し、
2. その速度にしたがって物質の量を増減させる、

といった一連の処理を書く。FluxReactorを継承している場合は、process()メソッドを用いて(2)の処理を簡単に行うことができる。process()メソッドの引数は、1秒あたりの反応ぶんしすうで、Float型である。

リアクターが扱うアクティビティーはステップ当たりの値であるが、リアクターウィンドウなどのユーザーインターフェースが表示するアクティビティーは1秒間当たりの値である。

KEYWORD	.tex	.c	.h	書式	必要
@CLASSNAME	○	○	○		○
@BASECLASS	○	○	○		○
@AUTHOR	○	△(*1)	△(*1)		
@EMAIL	○	△(*1)	△(*1)		
@DATE	○	△(*1)	△(*1)		
%VERSION	○	○	○		
@BRIEF_DESCRIPTION	○		○		
@DESCRIPTION	○			TEX	
@EQUATION	○			TEX	
%SUBSTANCE	○				○(*2)
%PARAMETER	○	○	○		○(*2)
@NOTES	○			TEX	
%INCLUDE_FILE_H			○		
@PRIVATE			○	C++	
@PROTECTED			○	C++	
@PUBLIC			○	C++	
%INCLUDE_FILE_C	○				

@INITIALIZE_FUNC	○	C++	○
@REACT_FUNC	○	C++	○
@OPTION_FUNC	○	C++	

表1 キーワードのまとめ(参照されるファイル、及び書式)

(\*1) △は、コメント分としてのみ参照される

(\*2) Reactorによっては必要のないものもある

記述法	意味
N_A	アボガドロ数
supersystem() -> stepper() -> deltaT()	ステップ幅
numSubstrate()	Substrateの数
numProduct()	Productの数
numCatalyst()	Catalystの数
numEffector()	Effectorの数
substrate(i) -> coefficient()	i番目のSubstrateの化学量論係数
substrate(i) -> concentration()	i番目のSubstrateの濃度 (単位 : M)
substrate(i) -> quantity()	i番目のSubstrateの分子数
substrate(i) -> substance(). supersystem() -> volume()	i番目のSubstrateのSystemの体積
product(i) -> coefficient()	i番目のProductの化学量論係数
product(i) -> concentration()	i番目のProductの濃度 (単位 : M)
product(i) -> quantity()	i番目のProductの分子数
product(i) -> substance(). supersystem() -> volume()	i番目のProductのSystemの体積
catalyst(i) -> concentration()	i番目のCatalystの濃度 (単位 : M)
catalyst(i) -> quantity()	i番目のCatalystの分子数
catalyst(i) -> substance(). supersystem() -> volume()	i番目のCatalystのSystemの体積
effector(i) -> concentration()	i番目のEffectorの濃度 (単位 : M)
effector(i) -> quantity()	i番目のEffectorの分子数
effector(i) -> substance(). supersystem() -> volume()	i番目のEffectorのSystemの体積

表2 メソッド一覧

## 2.2 Reactor Descriptionの例

以下に、既知の反応速度式に基づくRDファイルの書き方を例を用いて説明する。例えば、

$$v = \frac{(K_{cF}K_p[S] - K_{cR}K_s[P]) [E]}{K_s[P] + K_p[S] + K_sK_p}$$

という反応式に従うMichaelisUniUniReversibleReactorを作成するためのキーワードの記述は以下の通りになる。

```

@classname: MichaelisUniUniReversibleReactor
@baseclass: FluxReactor
@author: Kouichi Takahashi
@email: shafi@sfc.keio.ac.jp
@date: 1999 2/22

%version: ecs-v08, 0.1

@brief_description: Simple Henri-Michaelis-Menten UniUni Reversible kinetics.
@description: Simple Henri-Michaelis-Menten UniUni Reversible kinetics.
@equation: $$v=\frac{(K_{cF} K_p [S] - K_{cR} K_s [P]) [E]}{[K_s [P] + K_p [S] + K_s K_p]}$$

%substance: Substrate, 1, 1
%substance: Product, 1, 1
%substance: Catalyst, 1, 1
%substance: Effector, 0, 0

%parameter: Ks, Float, mol/l, Michaelis Constant of Substrate
%parameter: Kp, Float, mol/l, Michaelis Constant of Product
%parameter: KcF, Float, mol/l, Catalytic Constant (Forward)
%parameter: KcR, Float, mol/l, Catalytic Constant (Reverse)

@private: Float Ksp;

@initialize_func:
Ksp = Ks * Kp;
@react_func:
Float S = substrate(0)->concentration();
Float P = product(0)->concentration();
Float E = catalyst(0)->quantity();

Float velocity = (KcF * Kp * S - KcR * Ks * P) * E / (Ks * P + Kp * S + Ksp);
process(velocity);

```

Ks、Kpは共に定数であるために乗算(Ks \* Kp)の結果も常に一定である。よって、これは最初に一度計算すればよいので@PRIVATEで新たな定数 Kspを定義し、@INITIALIZE\_FUNCで Ksp = Ks \* Kp;

として計算する。

@REACT\_FUNC行内では、Substrateの濃度を表わすために、変数Sを定義して、substrate(0)->concentration()を代入する。

つまり記述は、

```
Float S=substrate(0)->concentration();
```

となる。同様に、PとEも定義する。

## 3 Reactor Source Code (.C .hファイル) への変換

RDファイルをE-CELL Systemにおける Reactorのソースコードに変換するプログラムは、 ``rd2ch'' である。このプログラムによって、 RDファイルからソースコードである.Cファイル及び.hファイルの2つができる。

このプログラムを次のように使う。

```
% rd2ch filename.rd
```

-dオプションで出力先のディレクトリを指定できる。

```
% rd2ch filename.rd -d output_directory
```

-tオプションでテンプレートファイルのあるディレクトリを指定する。

```
% rd2ch filename.rd -t template_directory
```

-hオプションでヘルプを表示する。

```
% rd2ch -h
```

## 4 Spec Sheetへの変換

RDファイルをReactor Spec Sheetに変換するプログラムは、 ``rd2tex'' である。

このプログラムの利用法を次のように使う。

```
% rd2tex filename.rd
```

このプログラムを実行すると、読み込まれたキーワードの数が表示されるようになっている。引数には複数のファイルが指定できる。この場合、それをまとめた内容のLaTeXファイルが outputされる。ファイル名にはワイルドカードの使用も可能であり、例えば、

```
% rd2tex *.rd
```

を実行すると、カレントディレクトリにある全てのRDファイルをまとめた Reactorのリストが作成される。

引数が複数の場合で、出力ファイル名を *filename.tex* にしたい場合に、 -oオプションを用いる。このオプションをつけない場合は自動的に *ReactorDescription.tex* というファイルが作成される。

```
% rd2tex filename1.rd filename2.rd ..... -o output_filename.tex
```

ヘルプメッセージを表示する。

```
% rd2tex -h
```

## 5 Makefileを用いたReactor Shared ObjectとSpec Sheetの自動作成

RDファイルから自動的にReactor Shared ObjectとReactor Spec Sheetを作成してくれる Makefileが用意されている。このMakefileを利用するには、 MakefileとRDファイルと同じディレクトリに置き、 Reactor

Shared Objectを作成する際には、

```
% make so
```

と入力する。実行したマシンのCPUの種類により、異なる名前のディレクトリに作成される。この名前はUNIXの「uname -m」コマンドで表示される名前で、例えば、「i686」、「alpha」である。E-CELL Systemも実行するマシンの種類により Reactorを探す際に同じ名前のディレクトリを参照する。例えば、「/home/testuser/reactor」というディレクトリがリアクターパスに設定され、CPUの種類がi686ならば、「/home/testuser/reactor/i686」というディレクトリも検索される。

Reactor Spec Sheetを作成する際には.tex、.dvi、.psのいずれの形式のファイルを作成するかによって、以下のように入力する。

```
% make {tex | dvi | ps}
```

不要なファイルを削除してディレクトリの中身を綺麗にする場合には、次のように入力する。

```
% make clean
```

## 6 ecellの実行

E-CELLが正常に実行されるためには、以下のことに注意する必要がある。ルールで指定されたReactor (.soファイル) は、リアクターパス(1章2.1節、4.5節参照)で指定するディレクトリに置く。Reactorが存在しない場合、E-CELLがエラーメッセージを表示する。

## 7 E-CELL Projectが標準で用意するReactor

Reactor Classname	Reversible/Irreversible Reaction
ZeroReactor	零次反応
MassActionReactor	素反応
MichaelisUniUniReactor	MichaelisMenten式に
MichaelisUniUniReversibleReactor	R MichaelisMenten式に
RapidEquilibriumReactor	R 迅速平衡を表す
ConstantParameterReactor	定数を表示する
OrderedUniBiReactor	R
OrderedBiReactor	
OrderedBiUniReactor	R
OrderedBiBiReactor	R
PingPongBiReactor	
PingPongBiBiReactor	R
RandomUniBiReactor	R
RandomBiUniReactor	R
RandomBiReactor	
RandomBiBiReactor	R
IsoUniUniReactor	R
IsoOrderedBiBiReactor	R
IsoTetraUniPingPongReactor	R
UniUnIisoUniUniPingPongReactor	R

Di-IsoPingPongReactor

R

表3 Standard Reactor

E-CELL User's Manual  
e Last Update \$Date: 2000/09/29  
10:31:10 \$

3章 Reactorの作成 [ 1章 : 2章 ]  
[ トップ ]